HITACHI PROGRAMMABLE CONTROLLER

EH-150 EHV-CPU

APPLICATION MANUAL
for NETWORK

NJI-491A(X)

Revision History

No.	Description of Revision	Date of Revision	Manual number
1	The first edition	2006.04	—
2	- Add explanation for EHV-CPU64 / 32 / 16. - Add explanation for Communication function. Revised chapter: Chapter 2	2007.02	NJI-491A(X)

Table of Contents

Chapter 1 Network Configuration 1-1 to 1-9

1.1 Communication port for CPU module 1-2
1.2 Network configuration for Communication module 1-5

Chapter 2 Specification of Communication port for CPU module 2-1 to 2-52

2.1 Features 2-1
2.2 Ethernet communication port 2-3
2.2.1 Task code communication port 2-3
2.2.2 ASR communication port 2-4
2.2.3 Reset function for Ethernet communication port 2-29
2.2.4 NTP client function 2-31
2.2.5 Factory setting 2-34
2.3 Serial communication port 2-35
2.3.1 Pin arrangement 2-35
2.3.2 Dedicated port 2-37
2.3.3 General-purpose port 2-39
2.3.4 Modem connection function 2-46
2.3.5 Connection between Serial communication port and Peripheral device 2-50
2.3.6 Connection method for RS-422 / 485 communication 2-51
2.4 USB communication port 2-52

Appendix 1 Cable Connection Diagram A-1

MEMO

Chapter 1 Network Configuration

EHV-CPU can configure various network systems depending on a combination of the communication port for CPU module and the communication module.

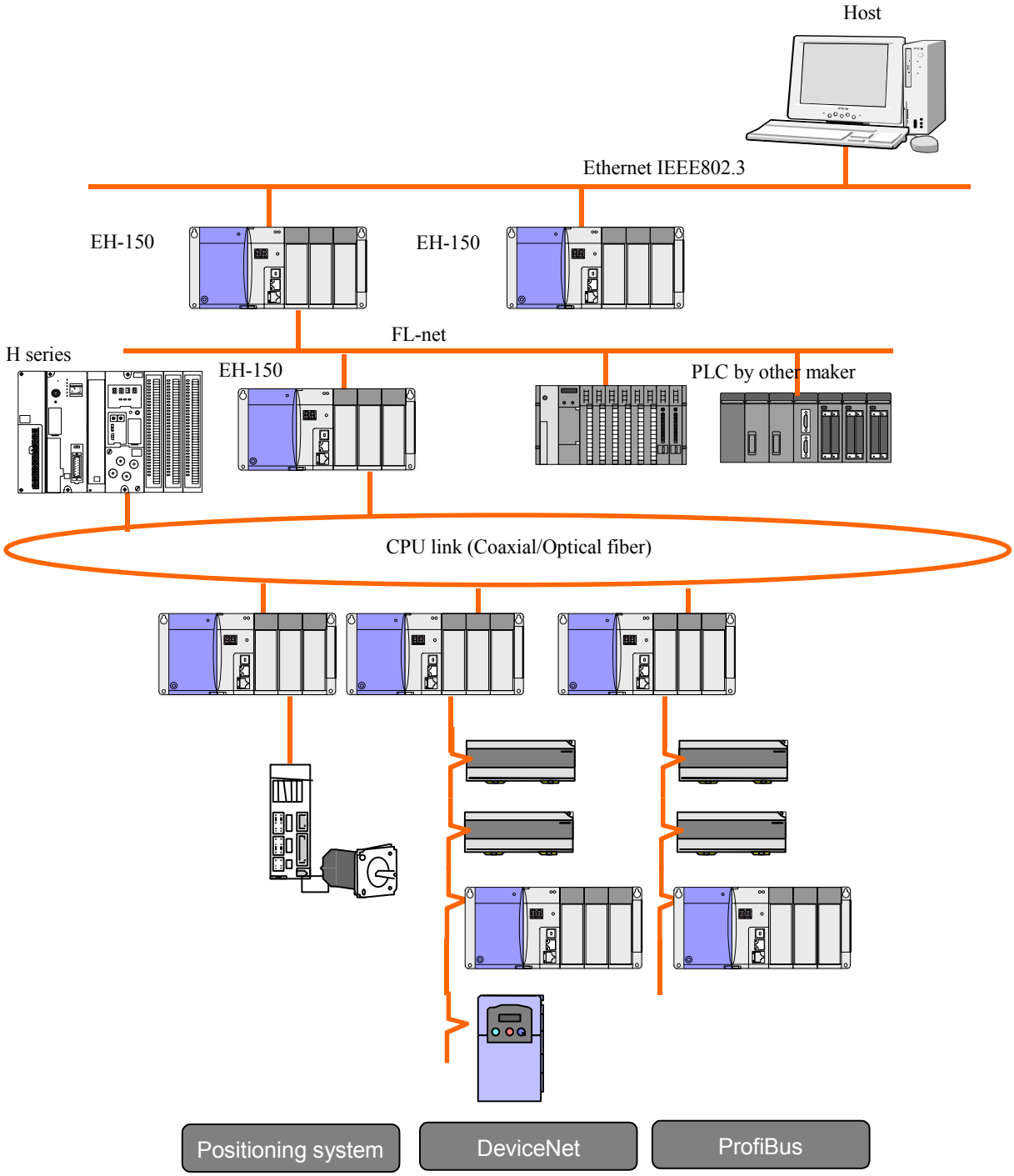
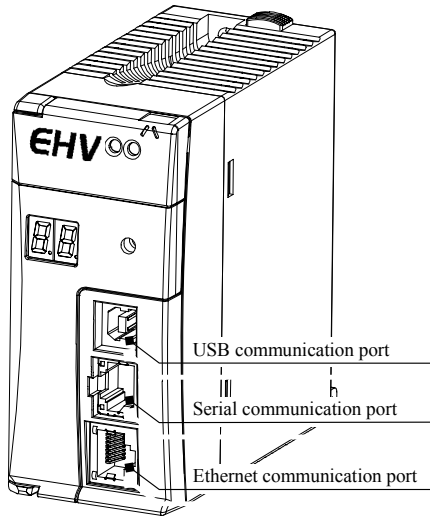


Figure 1.1 Network configuration for EHV-CPU

1.1 Communication port for CPU module

EHV-CPU is equipped with USB communication port, Serial communication port, and Ethernet communication port. The personal computer can be connected to every port, and you can create programs and monitor the system by using Control Editor which is a programming software.



USB communication port

It is a maintenance port for programming software.
Programming software can be used, connecting a notebook which is not equipped with a RS-232C serial port.

Serial communication port

RS-232C/422/485 can be switched like CPU currently in use.
Supports a dedicated procedure and a general-purpose communication.

Ethernet communication port

It has functions equivalent to the Ethernet communication module (4 dedicated procedure connections and 6 message communications). Support to network can be realized by the CPU module by oneself.

Figure 1.2 Communication ports for EHV-CPU

⚠ Caution

Please pay attention the following points on the communication port.

(1) If the Ethernet communication cable is connected to the serial communication port, the Ethernet communication port of the CPU module and external equipments connected to the Ethernet communication port may be damaged.

(2) A link with a network HUB etc. may not be established on the high speed communication by 100BASE-TX connection (100Mbps) or the link may be easy to cut because communication error occurs under the influence of installation environment, cable length, and noise. In these cases, please construct the network system with the following solutions.

(a) Increases the number of times to retry if necessary, using the TCP/IP communication for the protocol to communicate to other unit.

(b) Change the Ethernet communication speed to 10Mbps.

(3) If the programming tool and the USB communication port are used for connection, the programming tool may generate communication error under noise environment. If communication error is generated under noise environment, the serial port or the LAN port should be used for connection. And do not bring a communication cable close to other wiring and do not put the communication cable and the wiring into the same duct for the stable communication.

Reference

Since the Ethernet communication speed can be changed in Ver.x107 or newer, 5 types of communication speed (Auto-negotiation, 100M full-duplex/half-duplex, 10M full/duplex/half-duplex) can be set. The communication speed is set to 10M half-duplex at the shipment. (As for “x” in Ver.x107, it represents EHV-CPU128 when x is 0, it represents EHV-CPU64 when x is 1, it represents EHV-CPU32 when x is 2, and it represents EHV-CPU16 when x is 3. This version information is stored in the special internal output WRF050.)

Control Editor (Ver.2.00 or newer) or IP address setting tool can change the communication speed.

(1) Dedicated procedure communication (Task code communication)

A dedicated procedure communication of Hitachi PLC is called a task code communication. CPU can be controlled from the host and I/O can be read and written. Each sales maker provides a driver for this task code communication such as a touch panel and HMI (Human Machine Interface) software. For compatible Hitachi PLCs, it is unnecessary to create a special program.

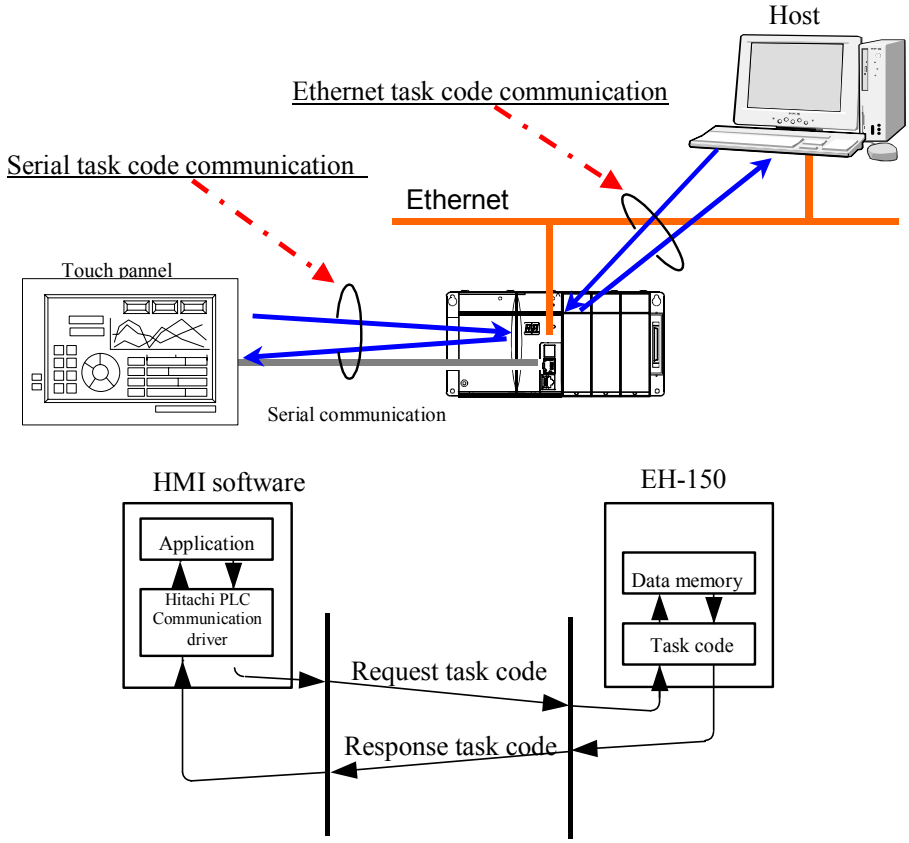


Figure 1.3 Task code communication

(2) No procedure communication

Serial communication port General-purpose communication

Serial communication port can be used as a general-purpose port which can be controlled by a user program. Various setting for communication and processing for transmitting and receiving can be created with the user program, matching to external equipments.

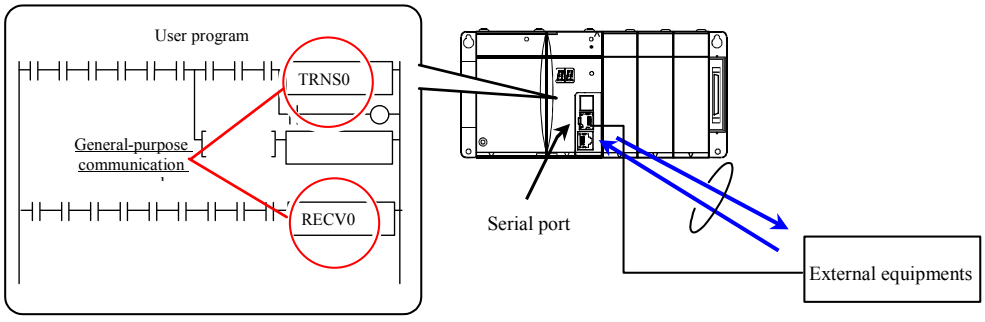


Figure 1.4 General-purpose communication for Serial communication port

Ethernet communication port ASR communication

ASR communication function can be used for the event transmitting function which transmits data from the CPU to the host actively at the event occurrence, the cycle transmission which transmits data to the host at constant interval, and when receiving message data from the host at any timing. There are 6 connections and the communication method can be specified respectively. Communication with the host is possible by only minimum setting.

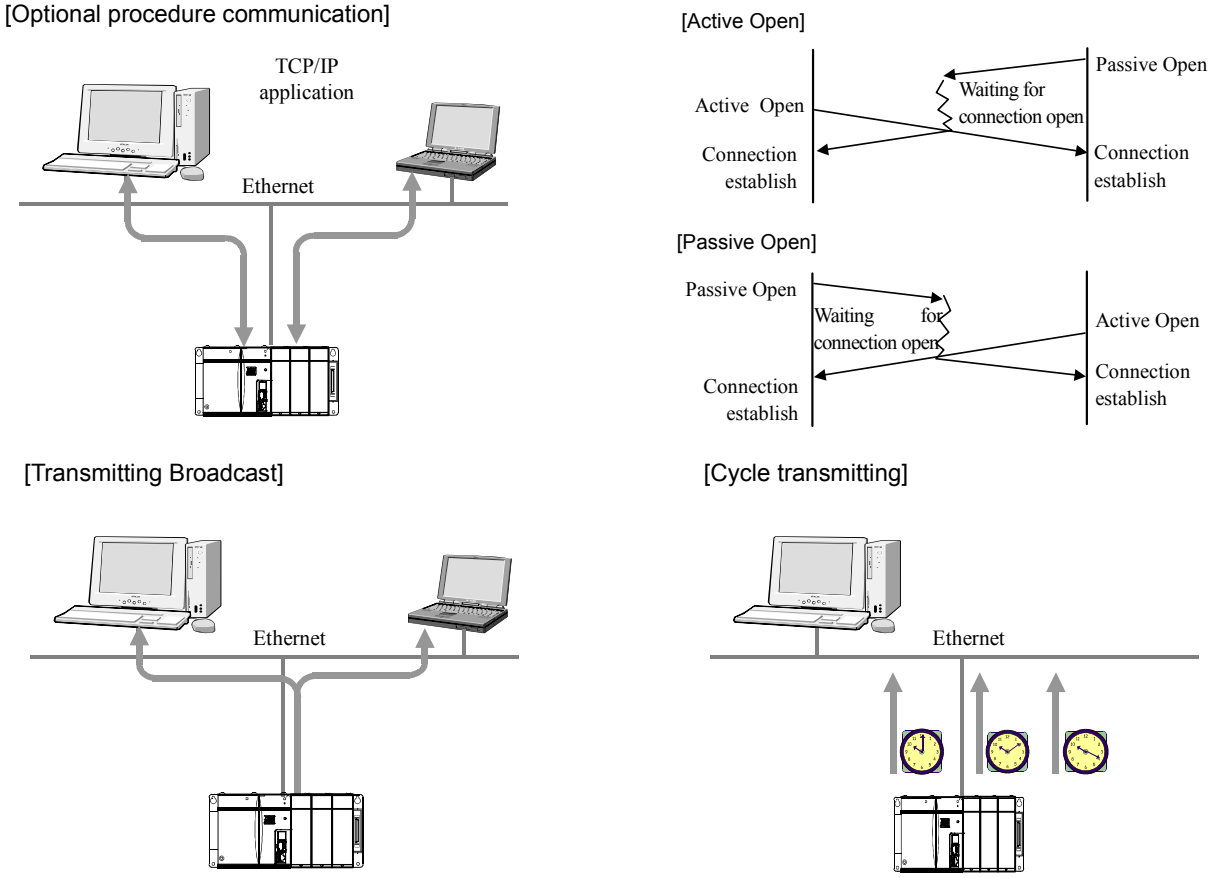


Figure 1.5 ASR communication for Ethernet communication port

Serial communication port Modem connection function

Serial communication port supports the modem connection function (only in setting RS-232C). If it is set so that the modem connection function can be used, necessary initial setting is performed to the modem automatically in connecting with the modem. If receiving from the modem is detected as an access from the host via commercial line, it becomes the status waiting for receiving task code.

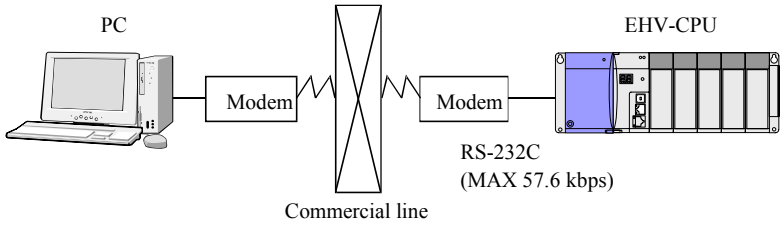


Figure 1.6 Modem connection function

1.2 Network configuration for Communication module

An example of a network configuration using the communication module is shown below. Refer to each instruction for detailed specification of each module.

(1) Ethernet module (EH-ETH)

If industrial equipments are connected to the information system network, it is useful for performing production control, system operation monitor, facilities monitor, and maintenance smoothly.

1.EH-ETH is mounted in a basic base of EH-150 system and is a communication interface module to connect the EH-150 system to Ethernet conformed to IEEE802.3.

2. EH-ETH connected to Ethernet functions as one station in the network.
And data can be exchanged between a personal computer and a server on the network.

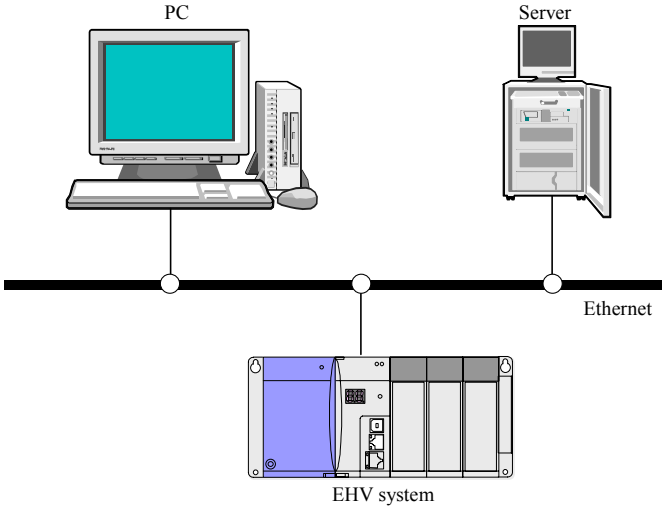


Figure 1.7 Example of Network configuration using Ethernet module

(a) 10 connections for transmitting and receiving of data can be used.
- There are 6 ASR connections for message communication and 4 connections for task code communication.
- Data can be transmitted and received by only one connection.
- One of TCP/IP and UDP/IP is selectable as a communication protocol which is used for each connection.
- Data up to 1454 bytes can be transmitted and received between PLC or the host.

(b) Simplifying a ladder program by **Web server function for communication setting** and **ASR function.**
- Various settings for starting communication are performed using a general-purpose Web browser. Setting information is stored on a built-in FLASH memory in EH-ETH as a file named “setup.dat”. This file is a text file format.
- Man-hour to create a ladder program can be reduced drastically by using ASR function.

(c) Programming is possible from Control Editor.
- Programming and I/O monitor via Ethernet of EH-ETH are possible by using the Control Editor. Maintenances of the program and the whole system improve by remote operation and monitor between PLC connected by Ethernet.

(2) DeviceNet™ Mater module (EH-RMD) / Slave controller (EH-IOCD)

Since DeviceNet™ master module / Slave controller conform to DeviceNet which is a open filed network, not only our master/slave device but also master/slave device made in other maker can be connected.

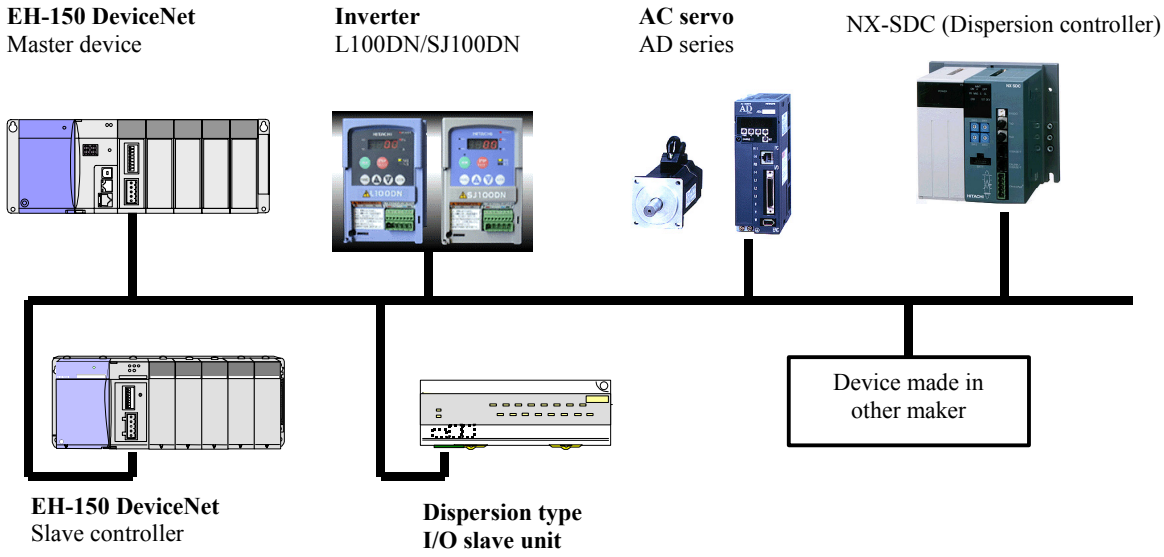


Figure 1.8 Example of Network configuration using DeviceNetTM Master module / Slave controller

DeviceNet Features				
Device type	Communication adaptor		Master/Scanner	Yes
Explicit peer-to-peer message		Yes	I/O slave message	
I/O peer-to-peer message		No	Bit strobe	Yes
Configuration consistency value		Yes	Polling	Yes
Fault node recovery		No	Cyclic	Yes
Communication speed	125/250/500 kbps		Change of state (COS)	Yes

1. Supports Link mode / Remote mode of EH-RMD.

Item	Link mode	Remote mode
Communication protocol	Conforms to DeviceNet release 2.0	
Support connection	Polling, Bit Strobe, Cyclic, COS, Explicit Message	
Number of installed units	2 units/CPU	4 units/CPU
Input and output point	256-word input 256-word output	1,024-point input and output
I/O assignment	CPU link	Remote 2
Configurator	Need	Need

2. Up to 16 modules can be mounted on EH-IOCD.
   EH-BS11A is not supported. Please use EH-BS3A/5A/8A.
3. EH-IOCD supports a digital I/O, an analog I/O module and a part of high-functional module.
4. Explicit message can be transmitted and received by a ladder program.

(3) Serial interface module (EH-SIO)

1. Communication with Serial communication equipments (General-purpose communication)

   User program performs communication with external equipments. “TRNS 9” which is a command for EH-SIO performs EH-SIO control by EHV-CPU and transmitting and receiving of data.

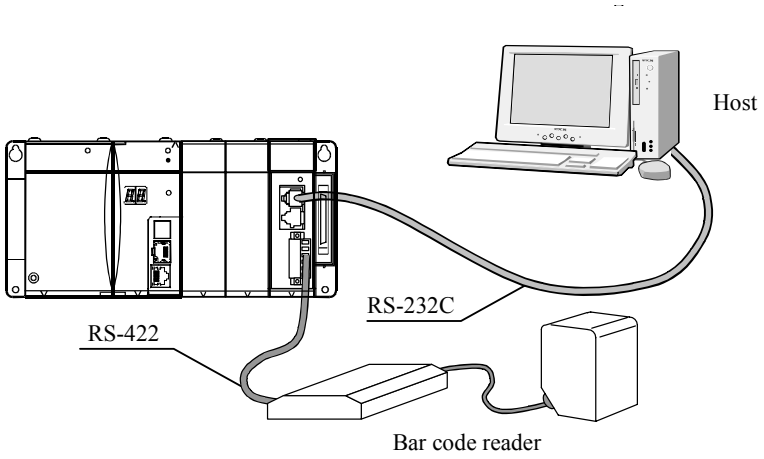


Figure 1.9 System configuration at general-purpose communication

2. Communication with Modbus protocol support equipments

   EH-SIO can communicate by Modbus protocol. Modbus slave equipments can be controlled with communication by setting EH-SIO to a master.

   And I/O of EHV-CPU can be accessed from the host by Modbus protocol.

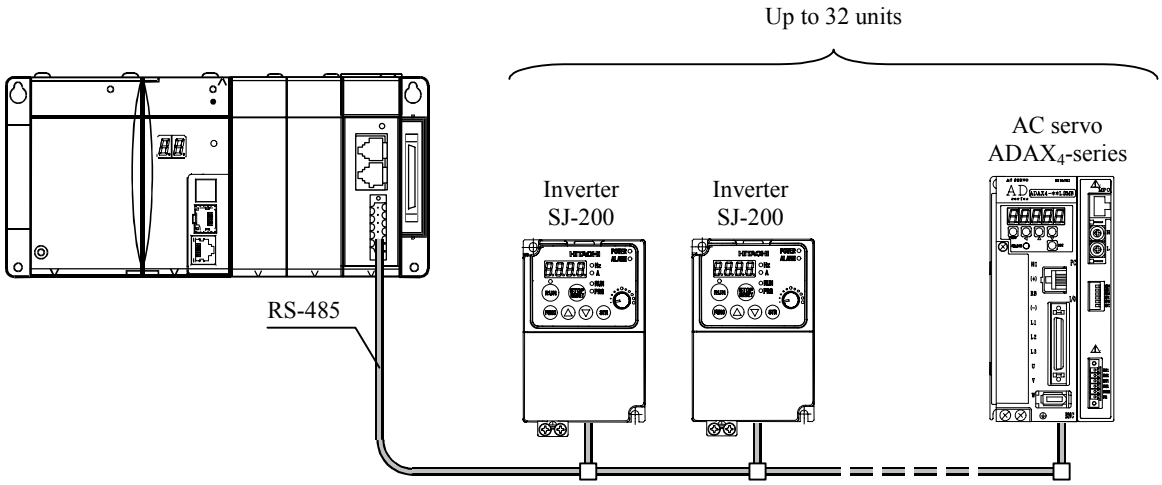


Figure 1.10 System configuration in controlling equipments supporting Modbus

3. Communication with Communication protocol (Hi-Protocol) support equipments for Hitachi H/EH series

   This can connect with HMI supporting Hi-Protocol.

   It can connect equipments supporting Hi-Protocol (HMI etc.), setting EH-SIO to Hi-Protocol mode.

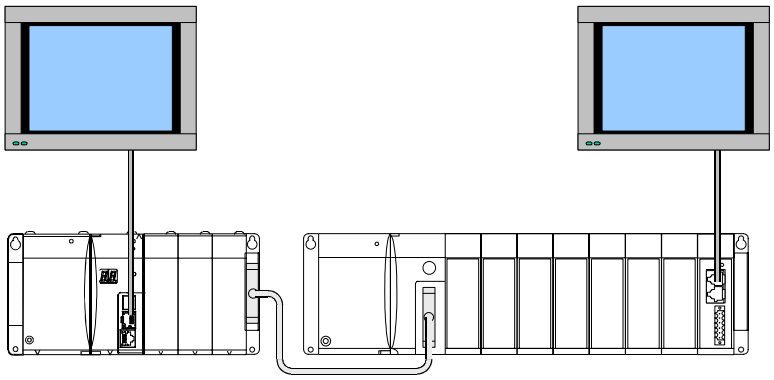


Figure 1.11 System configuration in connecting HMI

4. Simple data link function

   Simple data link is a function to exchange I/O information with a slave by communication, setting EH-SIO to a master.

   PLC which becomes a slave is EH-150*[1] and MICRO-EH*[1]. If initial setting of EH-SIO which becomes a master is completed, the I/O area is updated by the system automatically.

   *1 Models supporting a transmission control procedure 2 (with station No.) are applied.

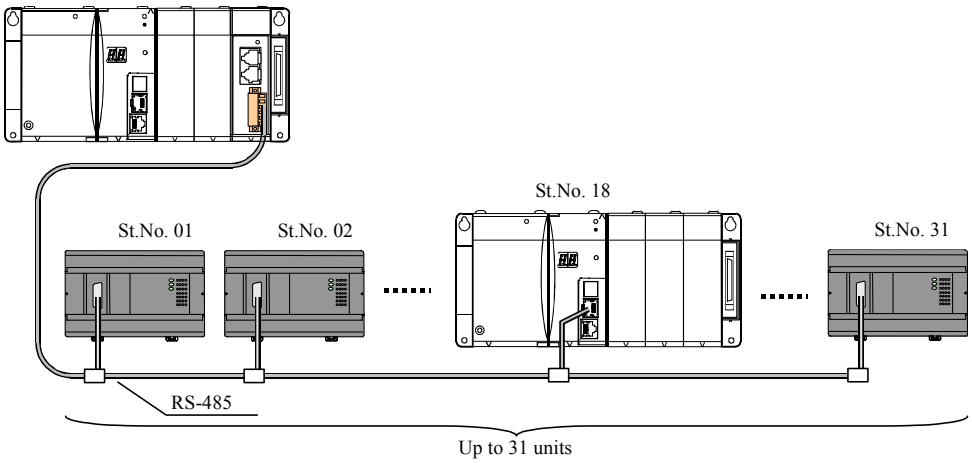


Figure 1.12 System configuration in simple data link

(4) CPU link module (Coaxial: EH-LNK, Optical: EH-OLNK)

CPU link can be formed by using the CPU link module (coaxial and optical).

And the entry into the existing H-series CPU link network is possible.

Fig.1.13 shows an example of a system configuration using the CPU link module (coaxial and optical).

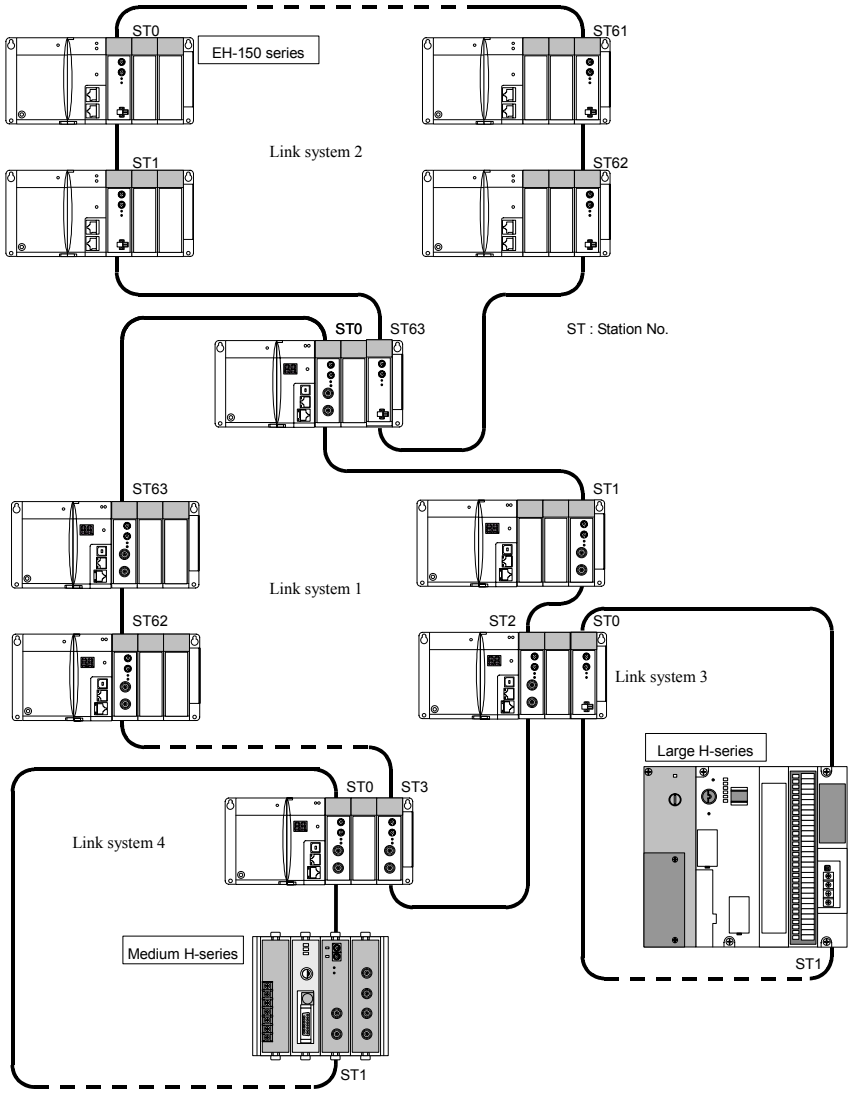


Figure 1.13 Example of System configuration for CPU link

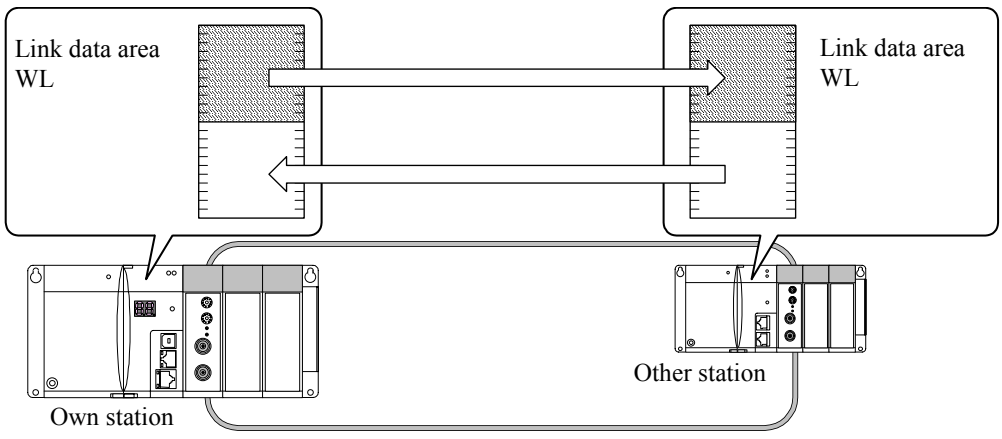


Figure 1.14 Outline of Link data area

MEMO

Chapter 2 Specification of Communication port for CPU module

2.1 Features

EHV-CPU has three communication ports as follows.

(1) Ethernet communication port

(2) Serial communication port

(3) USB communication port

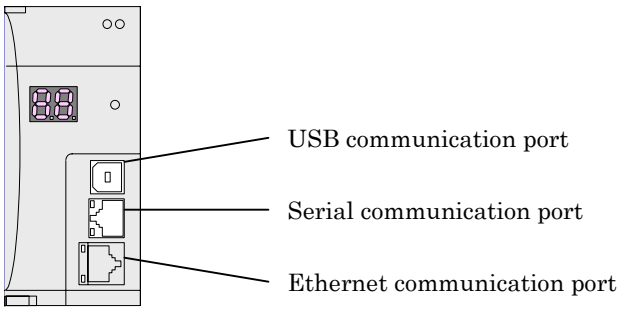


Figure 2.1 Communication port

(1) Ethernet communication port

EHV-CPU has 4 ports as a task code dedicated port for communicating by the dedicated protocol of H series. Programming and monitoring are possible by connecting the Control Editor. This port can also connect with a monitor available on the market corresponding to the H series dedicated protocol.

And since the ASR communication function is supported, transmitting and receiving message data are possible by the simple setting. Message data are transmitted at the event occurrence or periodically, and received automatically. Therefore, the network configuration matching to the system can be constructed.

In addition, since the SNTP client function is supported, the time information can be taken from the NTP server and the SNTP server on the network, and the time can be revised automatically. (Refer to NTP client function in this chapter for the SNTP client function.)

(2) Serial communication port

EHV-CPU supports RS-232C, RS-422, and RS-485 as a communication interface of the serial communication port. And also it supports a dedicated port for communicating by the H series dedicated protocol as the communication port and a general-purpose port which can control communication by a user program.

In the dedicated port, programming and monitoring are possible by connecting the Control Editor. This port can also connect with a monitor available on the market corresponding to the H series dedicated protocol. In addition, since the model connecting function is supported (only RS-232C), communication of the programming software via the model can be realized.

In the general-purpose port, since communication can be controlled by the user program, communication with external devices with the serial communication port is possible.

(3) USB communication port

USB communication port is a dedicated port to connect the Control Editor. Programming and monitoring are possible.

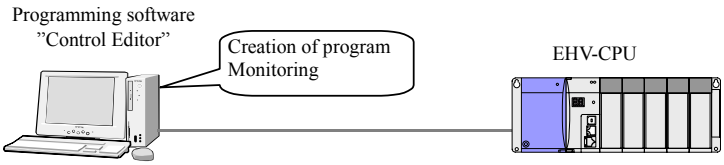


Fig.2.2 Programming software connection diagram

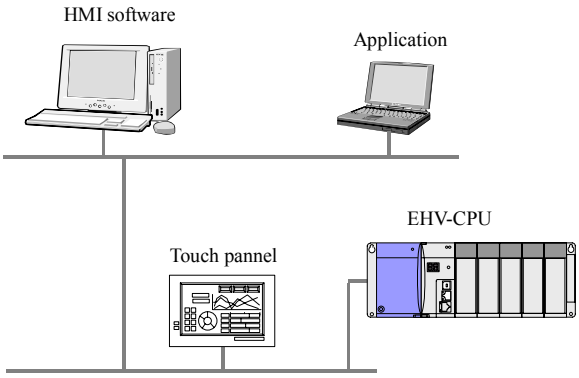


Figure 2.3 Example of Network configuration using dedicated protocol

Reference

Since the Ethernet communication speed can be changed in Ver.x107 or newer, 5 types of communication speed (Auto-negotiation, 100M full-duplex/half-duplex, 10M full-duplex/half-duplex) can be set. The communication speed is set to 10M half-duplex at the shipment. (As for "x" in Ver.x107, it represents EHV-CPU128 when x is 0, it represents EHV-CPU64 when x is 1, it represents EHV-CPU32 when x is 2, and it represents EHV-CPU16 when x is 3. This version information is stored in the special internal output WRF050.)

Control Editor (Ver.2.00 or newer) or IP address setting tool can change the Ethernet communication speed.

And set the Ethernet communication speed according to the following table.

Table 2.1 Task code communication specifications

			EHV-CPU				
			Auto-negotiation	100M		10M	
				Full-duplex	Half-duplex	Full-duplex	Half-duplex
Device to be communicated	Auto-negotiation		✓	—	✓	—	✓
	100M	Full-duplex	—	✓	—	—	—
		Half-duplex	✓	—	✓	—	—
	10M	Full-duplex	—	—	—	✓	—
		Half-duplex	✓	—	—	—	✓

2.2 Ethernet communication port

2.2.1 Task code communication port

Task code communication can achieve the following functions by combining individual communication command on a host program.

(1) CPU control (occupy / release, CPU status read, etc.)

(2) I/O control (all kinds of monitors)

(3) Memory write (all clear, batch transfer, etc.)

(4) Memory read (read of program, etc.)

(5) Response (all kinds of response from CPU)

This function can establish a system using the HMI software (SCADA, etc.) supporting HITACH H/EH series PLC Ethernet communication and a touch panel.

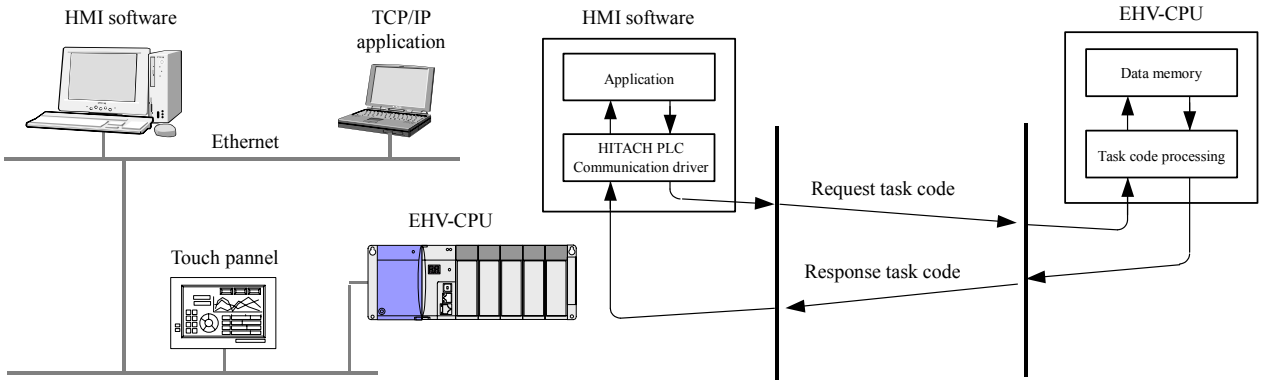


Figure 2.4 Composition of Task code communication equipment

Table 2.1 Specifications for task code communication

	Item	Specifications
1	Command system	HITACHI H/EH series PLC Ethernet task code (Server function)
2	Communication protocol	TCP/IP, UDP/IP
3	Logical port	Up to 4 (A port not to be used can be set up to the Not-Use.)
4	Logical port No.	Select any from 1,024 to 65,535
5	Timeout time	Invalid or Valid (Time between 1 to 65,535 sec. can be set up optionally.)

2.2.2 ASR communication port

ASR communication function can be used when message data is transmitted from this unit to the host actively at the event occurrence, and message data is received from the host at any time. And communication procedures can be established according to the system.

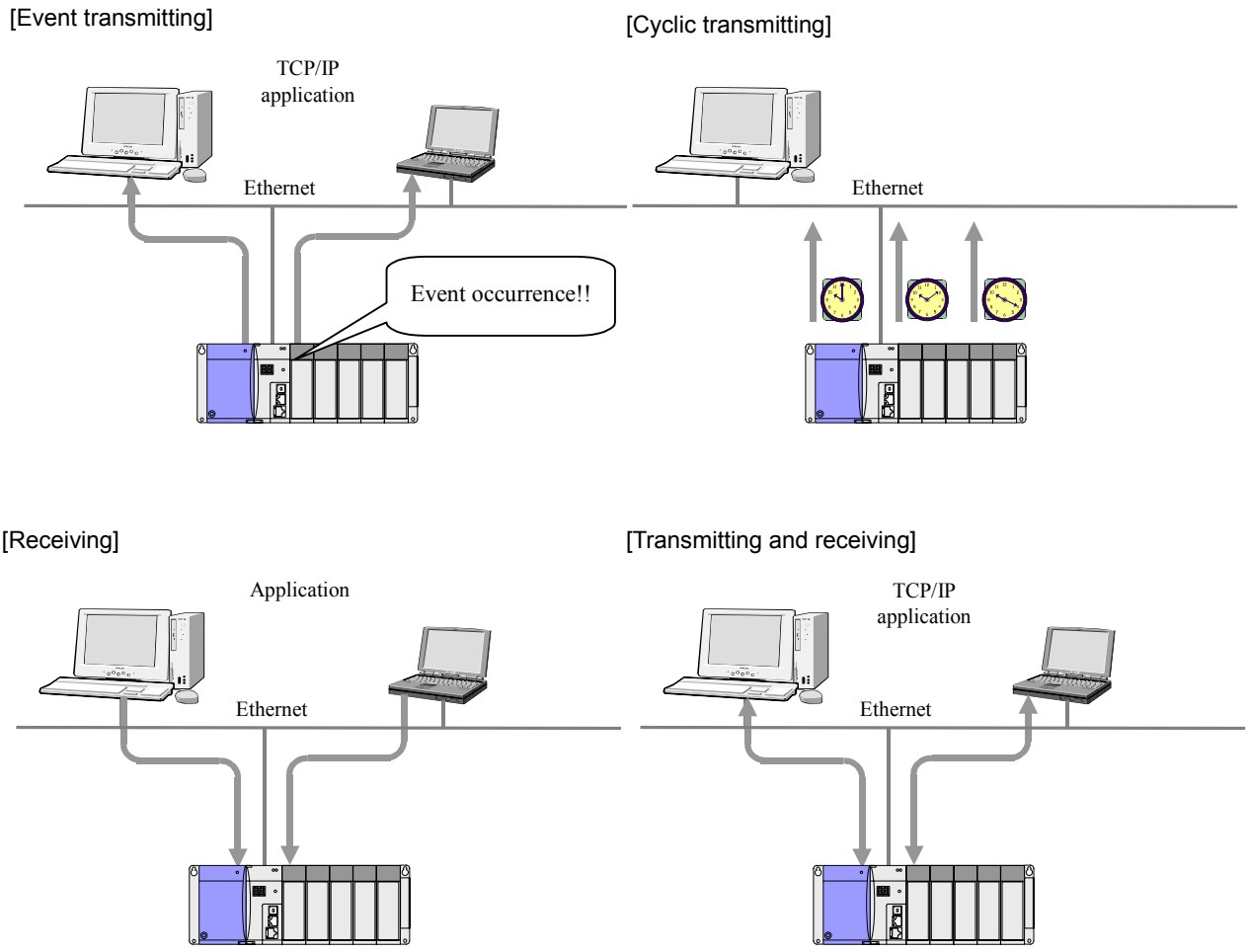


Figure 2.5 ASR communication port

Table 2.2 Communication specifications for ASR communication

	Item	Specifications
1	Communication protocol	TCP/IP, UDP/IP
2	Logical port	Up to 6 (A port not to be used can be set up to the Invalid.)
3	Maximum length of message	UP to 730 words
4	Transmitting area	Specifying from WX, WY, and internal output
5	Receiving area	Specifying from WY and internal output
6	Transmitting system	Event transmitting, Cyclic transmitting
7	Receiving system	Auto receiving

Communication type

You can specify the following 4 communication types.

Table 2.3 Communication type for ASR communication

	Type	Description
1	Not used	Not perform the transmitting and receiving
2	Transmitting and receiving	Performs the transmitting and receiving to the other station.
3	Only transmitting	Performs the transmitting to the other station only.
4	Only receiving	Performs the receiving only.

Connection type

You can specify the following 5 connection types.

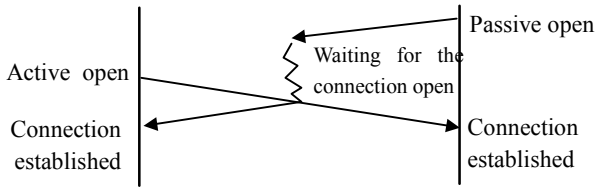
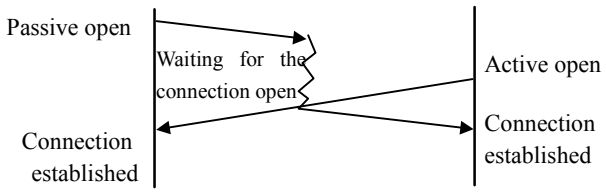
Table 2.4 Connection type for ASR communication

	Connection method
1	TCP/IP-Active open
2	TCP/IP-Passive open Specifies for the other station
3	TCP/IP-Passive open Optional for the other station
4	UDP/IP Specified for the other station
5	UDP/IP Optional for the other station

TCP/IP-Active open and TCP/IP-Passive open

When performing the ASR communication using TCP/IP, the logical transmission path for the connection with an open request should be established between EHV-CPU and the other station in advance. There are two methods to establish connection, the active open and the passive open.

Table 2.5 Connection method for ASR communication

No	Connection method	Description
1	Active Open	A method to establish connection by transmitting the open request later to the other station waiting for the connection open. 
2	Passive Open	A method to establish connection by receiving the open request from the other station, waiting for the connection open earlier. 

"Specified" and "Optional" for the other station

Message communication can be achieved with any station if TCP/IP-Active open is specified or UDP/IP-receiving is specified.

Transmitting Broadcast

When "Transmitting and receiving", or "Transmitting only" is specified using UDP/IP, message data can be exchanged between the logical ports which satisfy the following requirements.

(1) Nodes with the same network address (Multiple other stations)

(2) Nodes with the same logical port No., which can perform the UDP/IP communication (Multiple other stations)

(3) Nodes in status which can receive message (Multiple other stations)

This is called "Simultaneous transmission" or "Transmitting Broadcast".

【Transmitting Broadcast】

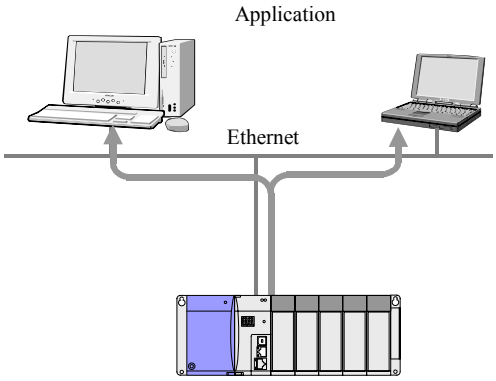


Figure 2.6 Transmitting Broadcast

Transmitting type

There are the following 2 transmitting types.

Table 2.6 Transmitting type for ASR communication

	Transmitting type	Description
1	Event transmitting	When the transmitting trigger bit specified is turned from OFF to ON, data in I/O memory specified as the transmitting area is transmitted To transmit data, the event transmission request flag should be ON for 120ms or longer and OFF for 120ms or more. Event transmitting request flag ON/OFF T_{OFF} T_{ON} T_{OFF}: Trigger bit OFF time (Min) T_{ON}: Trigger bit ON time (Min) T_{OFF}, T_{ON}≥120ms
2	Cyclic transmitting	Data in I/O memory specified as the transmitting area is transmitted at the interval (1 – 65,535 × 1sec, 1 – 65,535 × 40 ms) specified with the cyclic transmitting timer in a constant cycle.

Setup item

Items need to be set up depending on the combination of the communication type, the connection type, and the transmitting type. Required items are shown below. "✓" is marked to the item which should be set the parameter specifying in the following table for the communication.

* The Control Editor is used in setting up. When the port supply is turned on at next, the set information becomes effective.

Table 2.7 Setup items for ASR communication

	Communicatin type	Connection type	Transmitting type	Items which shoeld be setup											
				A	B	C	D	E	F	G	H	I	J	K	L
1	Transmitting and receiving	TCP/IP-Active	Event transmitting	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓
			Cyclic transmitting	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		TCP/IP-Passive Specified	Event transmitting	✓	✓	✓		✓	✓	✓	✓	✓	✓		
			Cyclic transmitting	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		
		TCP/IP-Passive Optional	Event transmitting	✓				✓	✓	✓	✓	✓	✓		
			Cyclic transmitting	✓			✓	✓	✓	✓	✓	✓	✓		
		UDP/IP Specified	Event transmitting	✓	✓	✓		✓	✓	✓	✓	✓	✓		
			Cyclic transmitting	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		
		UDP/IP Optional	Event transmitting	✓		✓		✓	✓	✓	✓	✓	✓		
			Cyclic transmitting	✓		✓	✓	✓	✓	✓	✓	✓	✓		
2	Only transmitting	TCP/IP active open	Event transmitting	✓	✓	✓		✓	✓	✓				✓	✓
			Cyclic transmitting	✓	✓	✓	✓	✓	✓	✓				✓	✓
		TCP/IP-Passice Specified	Event transmitting	✓	✓	✓		✓	✓	✓					
			Cyclic transmitting	✓	✓	✓	✓	✓	✓	✓					
		TCP/IP-Passice Optional	Event transmitting	✓				✓	✓	✓					
			Cyclic transmitting	✓			✓	✓	✓	✓					
		UDP/IP Specified	Event transmitting	✓	✓	✓		✓	✓	✓					
			Cyclic transmitting	✓	✓	✓	✓	✓	✓	✓					
		UDP/IP Optional	Event transmitting	✓		✓		✓	✓	✓					
			Cyclic transmitting	✓		✓	✓	✓	✓	✓					
3	Only receiving	TCP/IP-Active	—	✓	✓	✓					✓	✓	✓	✓	✓
		TCP/IP-Passice Specified	—	✓	✓	✓					✓	✓	✓		
		TCP/IP-Passice Optional	—	✓							✓	✓	✓		
		UDP/IP Specified	—	✓	✓	✓					✓	✓	✓		
		UDP/IP Optional	—	✓		✓					✓	✓	✓		

[A] Master station port No. [B] Other station IP address [C] Other station port No.

[D] Transmitting cycle timer [E] Transmitting area I/O type [F] Head I/O address in transmitting area

[G] Transmitting area size [H] Receiving area I/O type [I] Head I/O address in receiving area

[J] Receiving area size [K] Number of retry times [L] Retry interval

Transmitting area and receiving are information

In ASR communication, areas for the I/O memory which store message data to transmit and which store message data received need to be specifies.

Both the size which can be transmitted and received are 1 to 730 words.

Table 2.8 Transmitting and receiving are information for ASR communication

	Type		I/O memory area which can be specified		Number of areas which can be specified
			EHV-CPU128	EHV-CPU64 / 32 / 16	
1	Transmitting area information	WX	(*1)		10 areas (*2)
		WY	(*1)		
		WEX	(*1)		
		WEY	(*1)		
		WR	0 to FFFF		
		WL	0 to 3FF 1000 to 13FF 2000 to 23FF 3000 to 33FF 4000 to 43FF 5000 to 53FF 6000 to 63FF 7000 to 73FF		
		WM	0 to 7FFF		
		WN	0 to 1FFFF	0 to 7FFF	
2	Receiving area information	WY	(*1)		1 area
		WEY	(*1)		
		WR	0 to FFFF		
		WL	0 to 3FF 1000 to 13FF 2000 to 23FF 3000 to 33FF 4000 to 43FF 5000 to 53FF 6000 to 63FF 7000 to 73FF		
		WM	0 to 7FFF		
		WN	0 to 1FFFF	0 to 7FFF	

*1: Depends on the I/O assignment.

*2: The size which can be transmitted is up to 730 words totally regardless the number of setting areas

Number of retry times and Retry interval for Connection open

When TCP/IP-active is specified, the number of retry times and the retry interval for the connection open when failing can be specified.

When there is no response to the packet including the SYN flag*[1] transmitted from the EHV-CPU, the retry is performed three times. And this process is repeated for the number of retry times. The internal between cycles which consist of three retries is specified as the retry interval.

The following is an example in case where the number of retry times is set to 3 and the retry interval is set to 10 seconds.

*1: SYN flag is a connection open request flag.

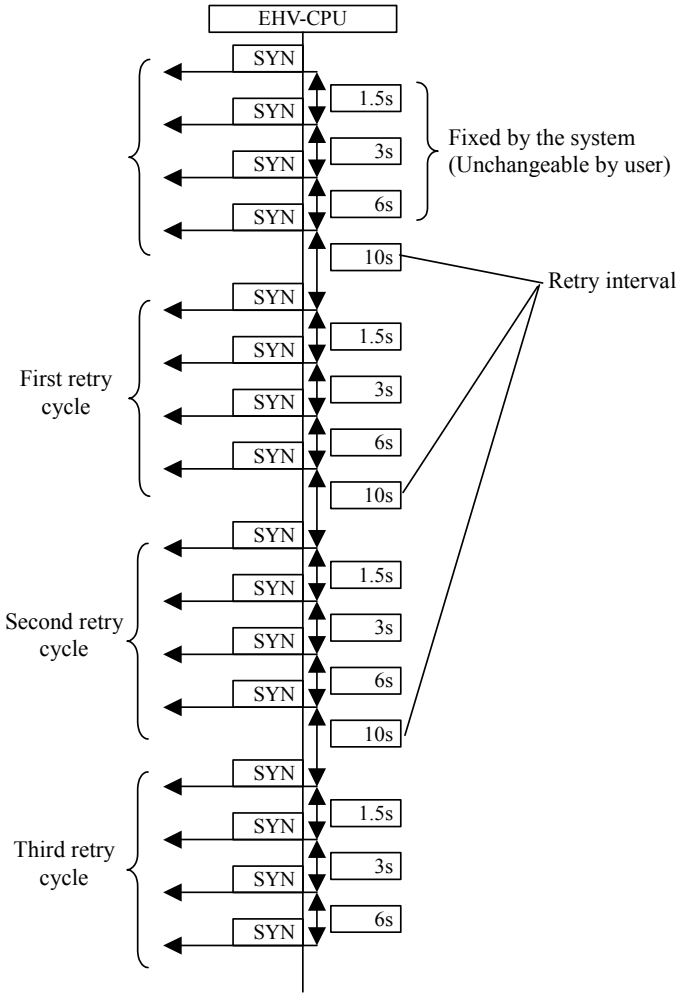


Figure 2.7 Connection open retry sequence

Status Register, Control register, Transmitting counter, Receiving counter

The status register, the control register, the transmitting counter, and the receiving counter for ASR communication are assigned to the special internal output WRF600 or later.

Table 2.9 Special internal output for ASR function

No.	Name		Set condition	Reset condition
WRF600	ASR port 1	Status register	Set by system	Clear system or user
WRF601		Control register	Set by user	Clear by user
WRF602		Transmitting counter	Set by system	Clear by user
WRF603		Receiving counter	Set by system	Clear by user
WRF604	ASR port 2	Status register	Set by system	Clear by system or user
WRF605		Control register	Set by user	Clear by user
WRF606		Transmitting counter	Set by system	Clear by user
WRF607		Receiving counter	Set by system	Clear by user
WRF608	ASR port 3	Status register	Set by system	Clear by system or user
WRF609		Control register	Set by user	Clear by user
WRF60A		Transmitting counter	Set by system	Clear by user
WRF60B		Receiving counter	Set by system	Clear by user
WRF60C	ASR port 4	Status register	Set by system	Clear by system or user
WRF60D		Control register	Set by user	Clear by user
WRF60E		Transmitting counter	Set by system	Clear by user
WRF60F		Receiving counter	Set by system	Clear by user
WRF610	ASR port 5	Status register	Set by system	Clear by system or user
WRF611		Control register	Set by user	Clear by user
WRF612		Transmitting counter	Set by system	Clear by user
WRF613		Receiving counter	Set by system	Clear by user
WRF614	ASR port 6	Status register	Set by system	Clear by system or user
WRF615		Control register	Set by user	Clear by user
WRF616		Transmitting counter	Set by system	Clear by user
WRF617		Receiving counter	Set by system	Clear by user

Details of the status register, the control register, the transmitting counter, and the receiving counter are described below.

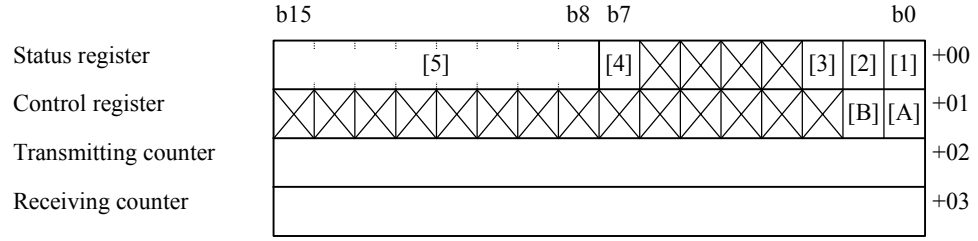


Figure 2.8 Status register, Control register, Transmitting counter, and Receiving counter

[Status register]

[1] ASR port status flag 1: Under open, 0: Under close

[2] Event transmitting completion flag 1: Transmitting completion

[3] Receiving completion flag 1: Receiving completion

[4] Error flag 1: Error occurrence

[5] Error code 0x01: Event transmitting request flag [B] is turned ON while ASR port status flag [A] is closed.

0x02: Event transmitting request flag is turned ON again in status which the transmitting of message is not completed.

[Control register]

[A] ASR port open request flag 1: Open request, 0: Close request

[B] Event transmitting request flag 1: Transmitting start

[Transmitting counter]

Stores the number of transmitting of message data.

[Receiving counter]

Stores the number of receiving of message data.

(1) ASR port status flag [1] and ASR port open request flag [A]

[TCP/IP Active]

If user turns on the ASR port open request flag [A], the system will open the connection with the communication other stations. If the other station is waiting for the connection open, the connection will be open normally and the ASR port status flag [1] will turn ON and it will be indicated that the connection is opening. If the other station is not waiting for the connection open and is not found, the connection will not be open normally and the ASR port status flag [1] is still OFF and it will be indicated that the connection is closing. If user turns off the ASR port open request flag [A] while the connection is opened, the connection will be closed and the connection status request flag [A] will turn OFF.
And if the other station closes the connection, user must turn off the ASR port open request flag [A] because it does not turn OFF.

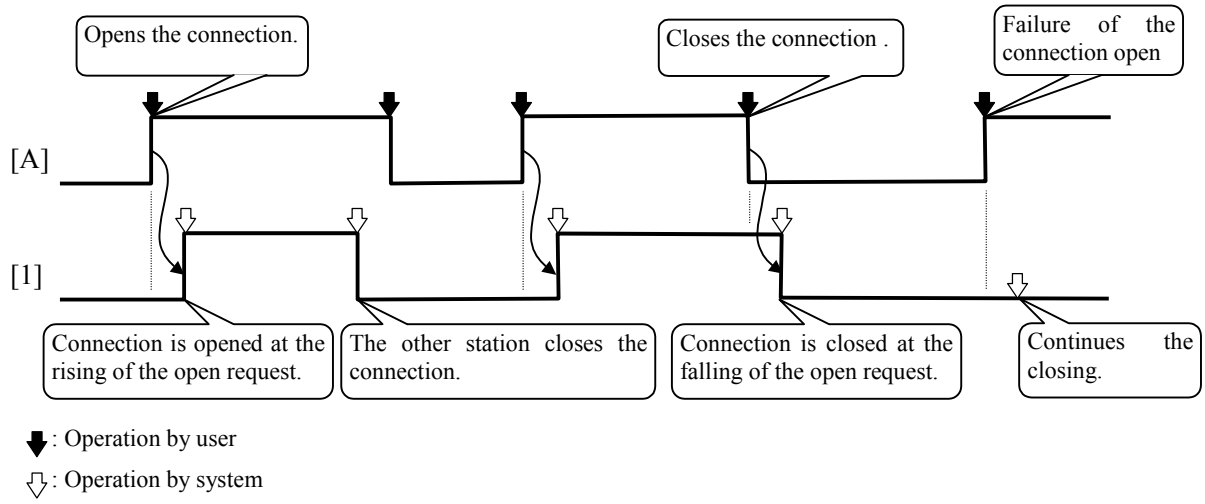


Figure 2.9 TCP/IP Active open for ASR port status flag and ASR port open request flag

[TCP/IP Passive]

If user turns on the ASR port open request flag [A], the connection open will become the waiting status. In this case, if the other station transmits the connection open request, the connection with the other station will be opened and the ASR port status flag [1] will turn ON and it is indicated that the connection is opening. The connection open does not become the waiting status when the ASR port open request flag [A] is OFF. In this case, if the other station transmits the connection open request, the connection with the other station will not be opened.

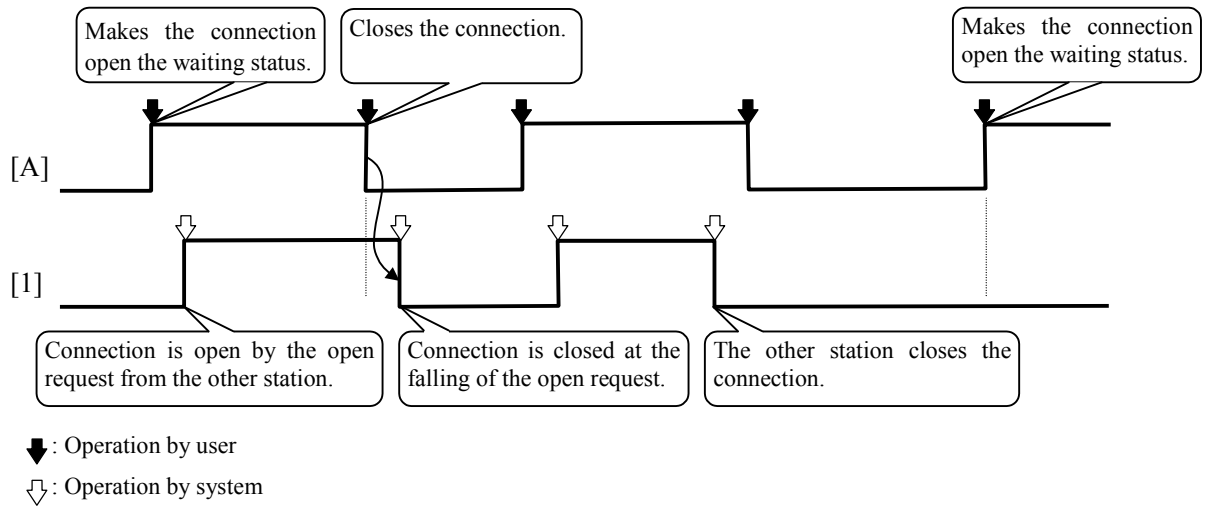


Figure 2.10 TCP/IP Passive open for ASR port status flag and ASR port open request flag

(*) In case where the connection is not closed normally.

When the connection is not closed normally for reasons that the cable came off, etc., the process mentioned above is not performed. When the set of the transmitting and receiving is specified to “Transmitting” or “Transmitting and receiving”, it is detected by the transmitting timeout that the connection is not opened normally and the ASR port status flag [1] is turned ON.

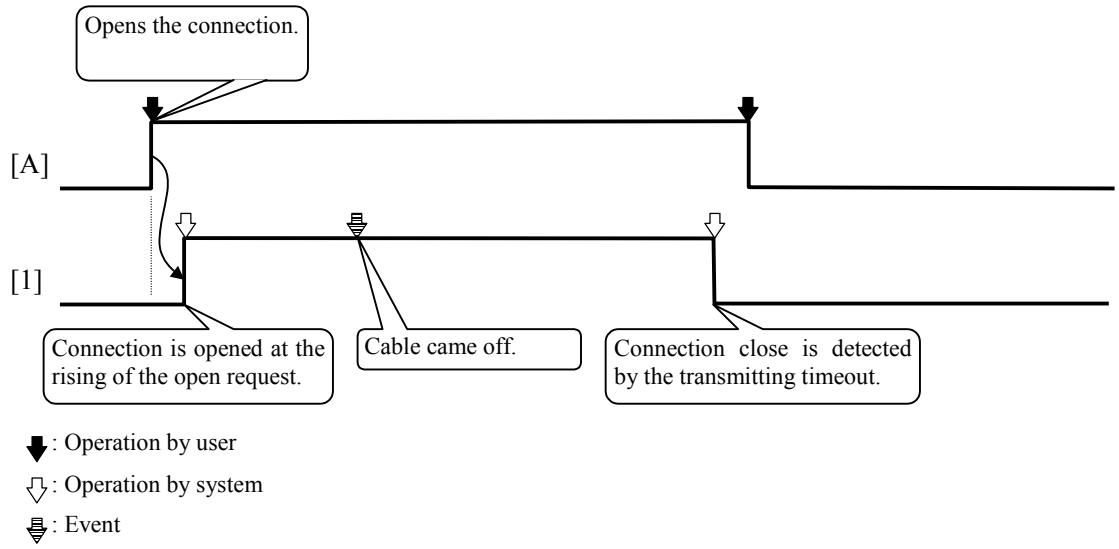


Figure 2.11 TCP/IP Transmitting and Transmitting and receiving side for ASR port status flag and ASR port open request flag

When the set of transmitting and receiving is specified to “Receiving”, it is not detected that the connection is closed because of waiting for message data from the other station. When the connection open request is transmitted from the other station again, it is detected that the connection is closed and the ASR port status flag [1] is turned OFF.

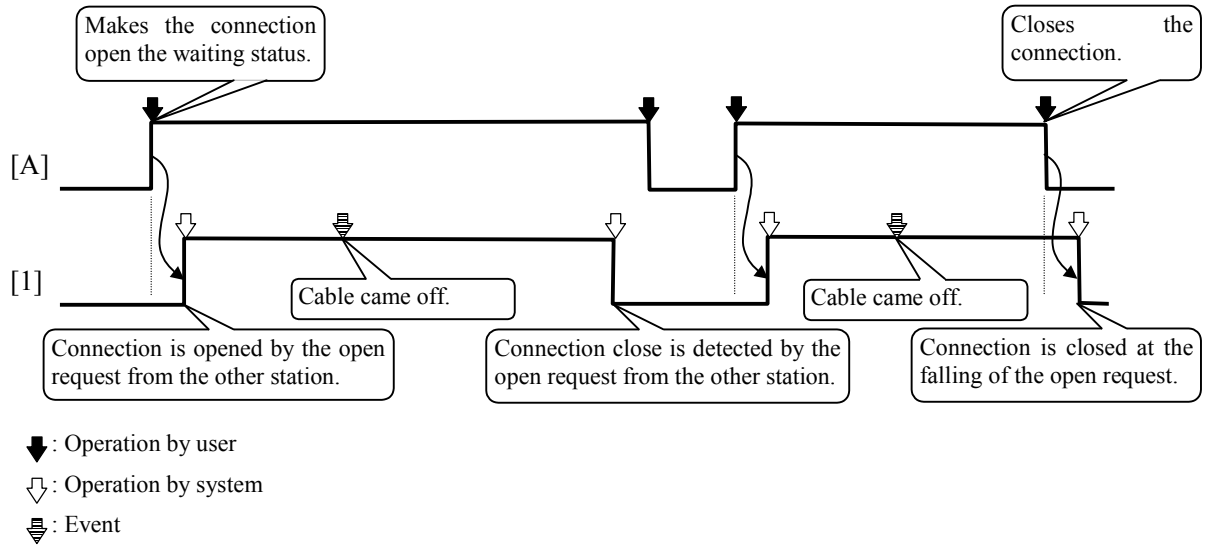


Figure 2.12 TCP/IP Receiving side for ASR port status flag and ASR port open request flag

[UDP/IP]

If user turns on the ASR port open request flag [A], the ASR port is opened and it is that the ASR port is opening because the ASR port status flag [1] is turned off. In this status, message data can be transmitted and received.

When UDP/IP other station is fixed, ARP packet is transmitted to communication destination when message data is transmitted for the first time. If there is no response to this ARP packet, ASR port is closed and the ASR port status flag [1] is turned off. In case of only receiving or when message data is transmitted after second time, the system does not close the ASR port because ARP packet is not transmitted to communication destination.

If user turns off the ASR port open request flag [A], the ASR port is closed and it is indicated that the ASR port is closing. In this status, message data cannot be transmitted and received.

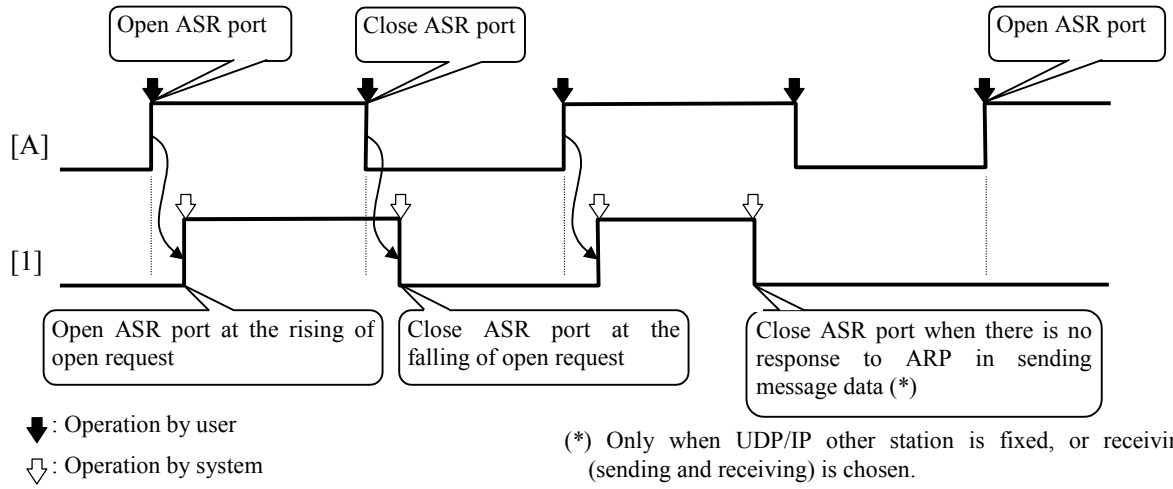


Fig. 2.13 UDP/IP for ASR port status flag and ASR port open request flag

(2) Event transmitting completion flag [2] and Event transmitting request flag [B]

If user turns on the event transmitting request flag [B] when message data can be transmitted and received (the connection is established in TCP/IP, and ASR port is opened in UDP/IP), the system will transmit message data. If the transmitting of message data is completed, the system will turn on the event transmitting completion flag [2]. If user performs the event transmitting again, user must turn on the event transmitting request flag [B] after turning OFF. And when monitoring whether the transmitting to the event transmitting request has been completed or not, user needs t turn off the event transmitting completion flag before turning on the event transmitting request flag.

If the event transmitting request flag [B] is turned ON when message data cannot be transmitted and received (the connection is not established in TCP/IP, and ASR port is not opened in UDP/IP), the error flag and the error code are set because of error.

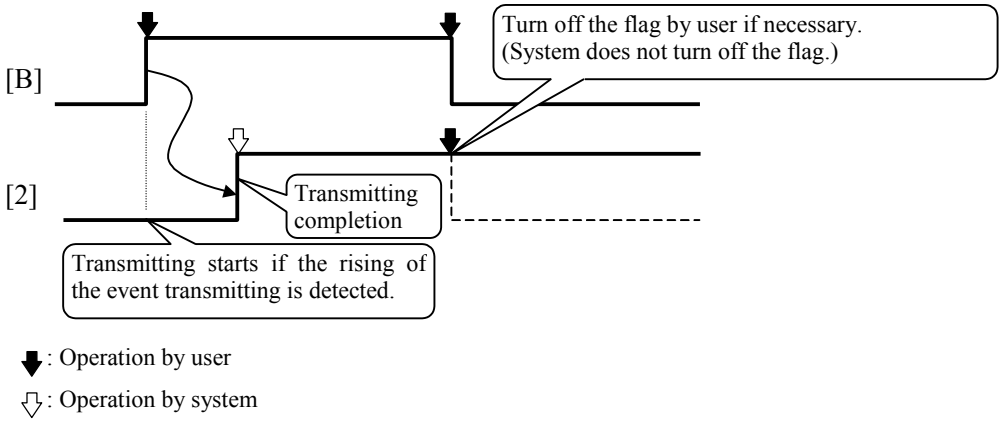


Figure 2.14 Event transmitting completion flag and Event transmitting request flag

(3) Receiving completion flag [3]

If the receiving of message data is completed, the receiving completion flag [3] will turn ON. Since the system turns on this flag whenever the receiving is completed, user needs to turn off this flag when monitoring the receiving using this flag.

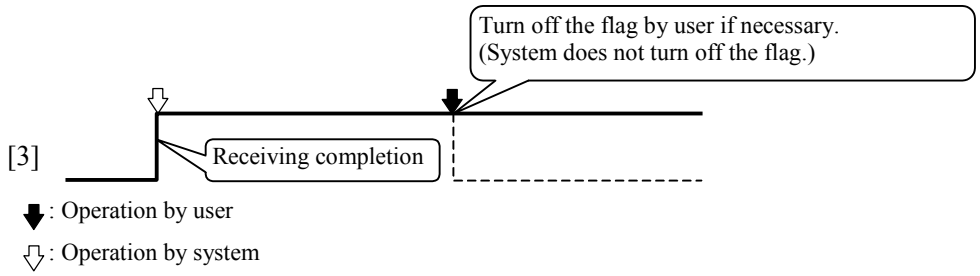


Figure 2.15 Receiving completion flag

(4) Error flag [4] and Error code [5]

The system will turn on the error flag [4] and store the error code [5] if factors of the error are found in the system. User must clear this flag and area if necessary because the system does not clear them.

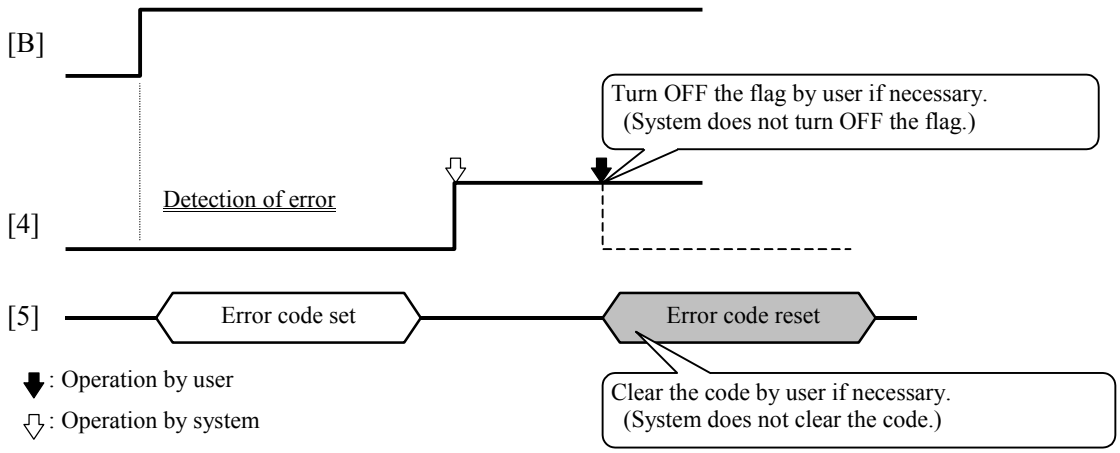


Figure 2.16 Error flag and Error code

(5) Transmitting counter and Receiving counter

The increment of the transmitting counter is performed when message data is transmitted.

The increment of the receiving counter is performed when message data is received.

User must clear the transmitting counter and the receiving counter if necessary because the system does not clear them.

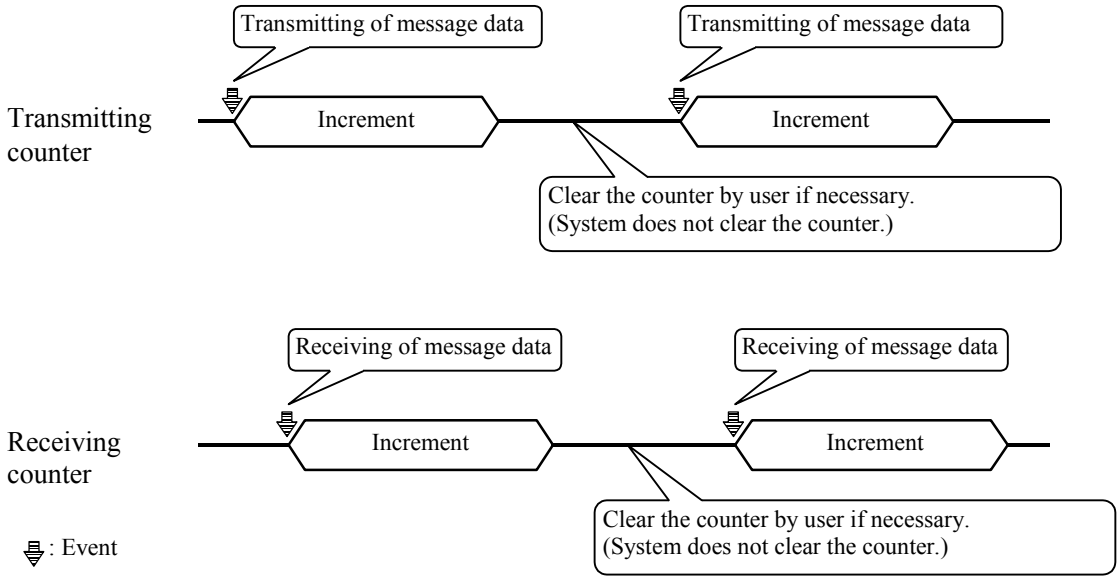


Figure 2.17 Transmitting counter and Receiving counter

TCP/IP protocols

When using TCP/IP to communicate, it is necessary to establish the connection between communication stations. Otherwise, message data cannot be sent and received. In order to establish the connection, set one side to the **TCP/IP connection active open**, and set the other side to the **TCP/IP connection passive open**.

Open a port of the TCP/IP connection passive open side, and then open a port of the TCP/IP connection active open after the connection open stood by to open. After the connection established, data can be sent and received.

And in order to close the connection, close the port you want to close first.

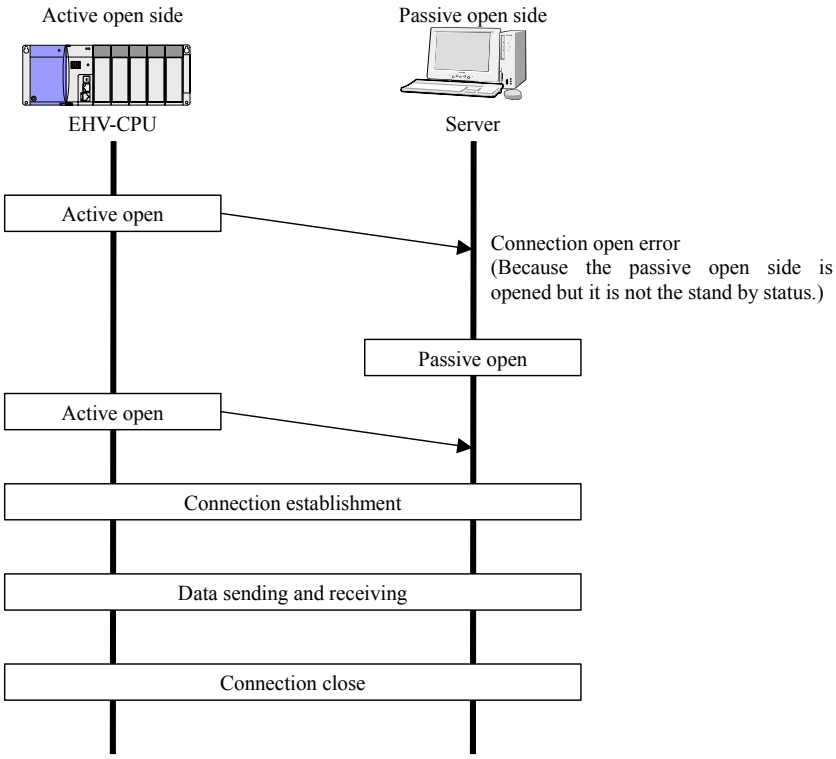


Figure 2.18 Example of TCP/IP protocols

Reference

In order to open the port on the ASR communication port, turn on the ASR port open request flag of the control register.

Ex.) In case of ASR communication port 1: WRF601 = H0001

In order to close the port on the ASR communication port, turn off the ASR port open request flag of the control register.

Ex.) In case of ASR communication port 1: WRF601 = H0000

UDP/IP protocols

In the ASR communication port, when using UDP/IP to communication, it is necessary to open the ASR port. Message data can be sent and received after the ASR port is opened. Message data cannot be sent and received if the ASR port is closed. When receiving messages while the ASR port is closed, the receiving data is cancelled.

In the ASR communication port, close the port to terminal the communication.

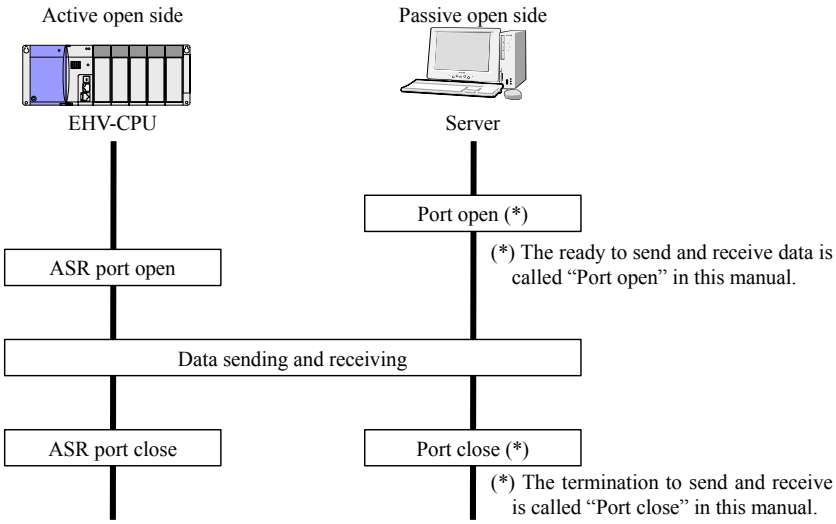


Figure 2.19 Example of UDP/IP protocols

Reference

In order to open the port on the ASR communication port, turn on the ASR port open request flag of the control register.

Ex.) In case of ASR communication port 1: WRF601 = H0001

In order to close the port on the ASR communication port, turn off the ASR port open request flag of the control register.

Ex.) In case of ASR communication port 1: WRF601 = H0000

Sample program

[Sample 1]

Network consists of two EHV-CPUs as follows, and the Control Editor sets the ASR communication. The setting information for two EHV-CPUs is as follows.

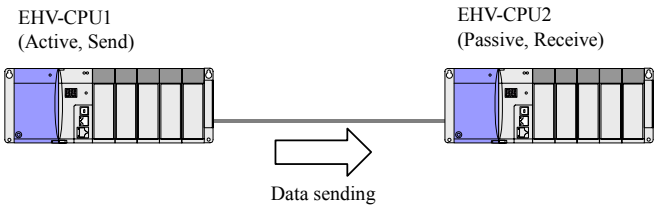


Figure 2.20 Connection diagram of Sample 1

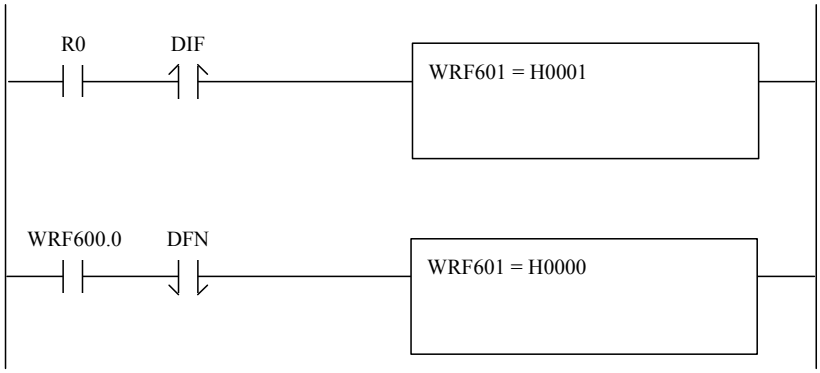
Table 2.10 Setting information of Sample 1

	Setting	EHV-CPU1	EHV-CPU2
1	IP address	192.168.0.1	192.168.0.2
2	Port No.	4000	4000
3	Protocol	TCP/IP-Active, Specified	TCP/IP-Passive, Specified
4	Send / Receive	Send	Receive
5	Access Point – IP address	192.168.0.2	192.168.0.1
6	Access Point – Port No.	4000	4000
7	Send Timing	Cyclic sending: 1 second	—
8	Transmission area	WR0 to WRF	—
9	Receiving area	—	WN0 to WNF

[Description of Sample program]

The connection is opened and the transmitting starts when R0 of EHV-CPU1 is turned ON.

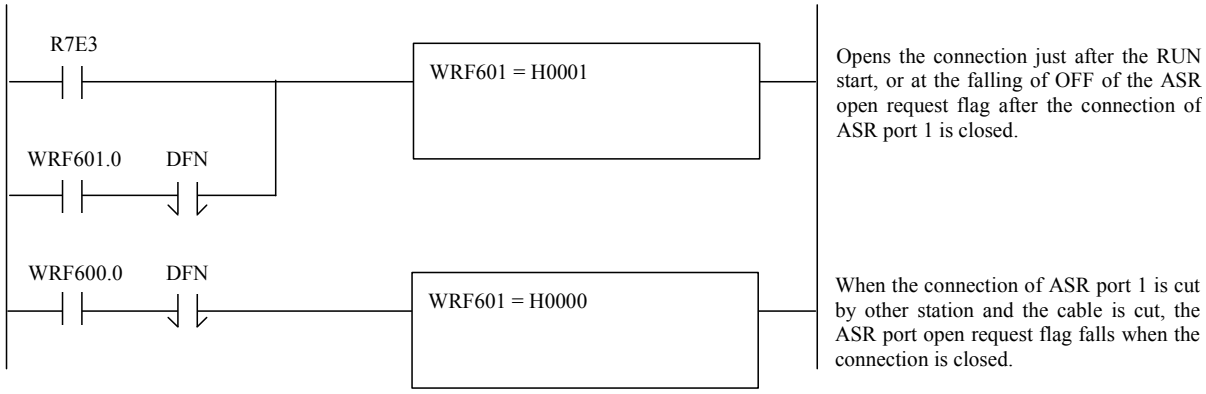
[Sample program of EHV-CPU1]



While the connection of ASR port a is not established, ASR port a is opened at the rising edge of user turns on R0. After the connection is established, ASR communication is started according to the setting and data is transmitted at one second interval.

When the connection of ASR port 1 is cut by other station and the cable is cut, the ASR port open request flag falls when the connection is closed.

[Sample program of EHV-CPU2]



[Sample 2]

Network consists of one EHV-CPU and one server as follows, and the Control Editor set the ASR communication of EHV-CPU. The setting information for EHV-CPU3 and the server is as follows.

When the even occurs, 5-word data from WRF00B to WRF00F of EHV-CPU3 is transmitted from EHV-Cpu3 to the server.

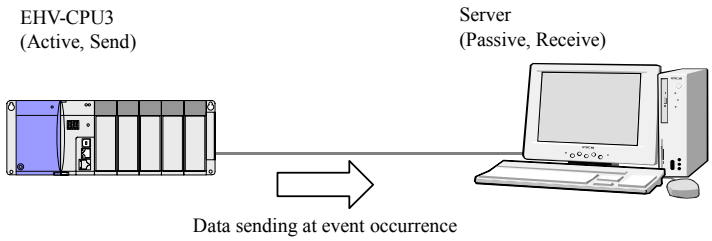


Figure 2.21 Connection diagram of Sample 2

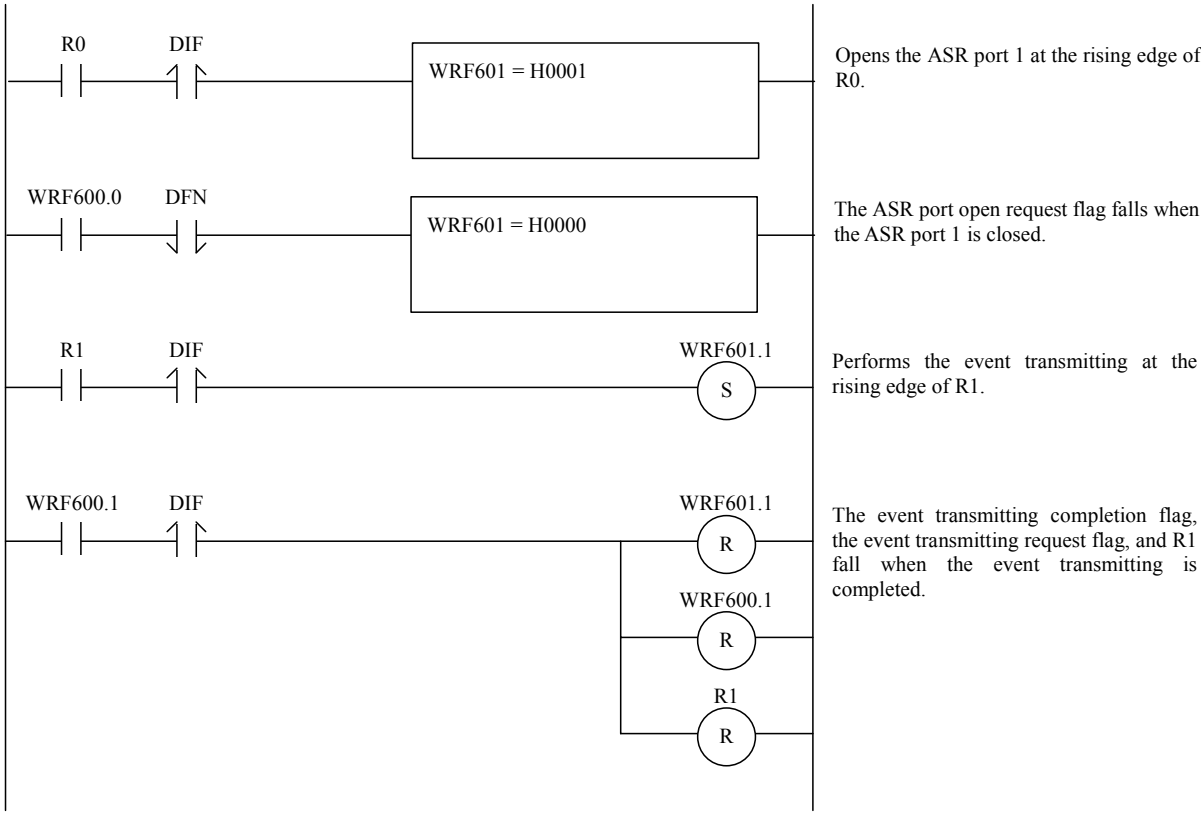
Table 2.11 Setting information of Sample 2

	Setting	EHV-CPU3	Server
1	IP address	192.168.0.10	192.168.0.11
2	Port No.	4001	4002
3	Protocol	TCP/IP-Active, Specified	TCP/IP-Passive, Optional
4	Send / Receive	Send	Receive
5	Access Point – IP address	192.168.0.11	—
6	Access Point – Port No.	4002	—
7	Send Timing	Event sending	—
8	Transmission area	WRF00B to WRF00F	—
9	Receiving area	—	—

[Description of sample program]

The connection is opened when R0 of EHV-CPU3 is turned ON, and the event transmitting is performed by turning ON R1 at the event occurrence.

[Sample program of EHV-CPU3]



[Sample 3]

Network consists of EHV-CPU and Web controller as follows, and the Control Editor sets the ASR communication of EHV-CPU. ASR communication of Web controller is set using the Web browser. The setting information of EHV-CPU4 and Web controller is as follows.

The connection is established between EHV-CPU4 and Web controller in two seconds later after RUN of EHV-CPU4, 16-word data from WR0 to WRF of EHV-CPU4 is transmitted from EHV-CPU to Web controller at one second interval. In Web controller, the received data is stored in WR0 to WRF. And 16-word data from WR10 to WR1F of Web controller is transmitted from Web controller to EHV-CPU4, and the received data is stored in WR10 to WR1F in EHV-CPU4.

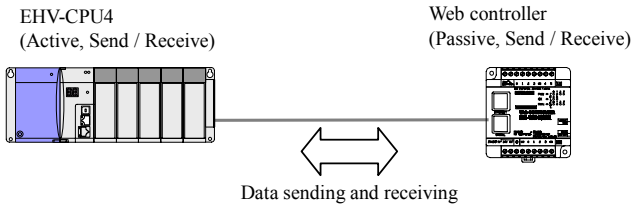


Figure 2.22 Connection diagram of Sample 3

Tale 2.12 Setting information of Sample 3

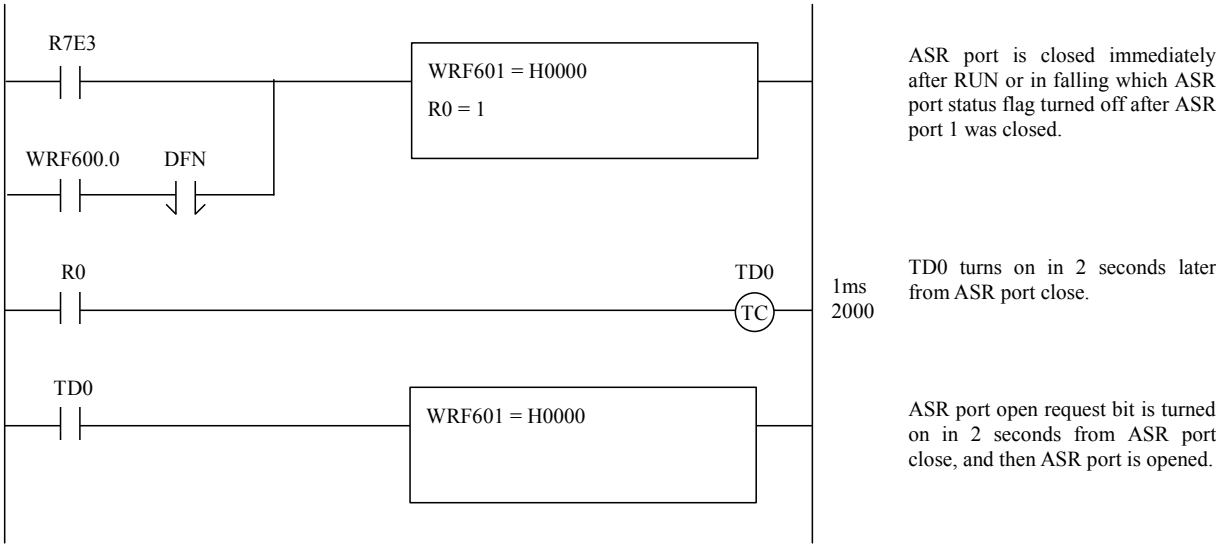
	Setting	EHV-CPU4	Web controller
1	IP address	192.168.0.1	192.168.0.2
2	Port No.	4000	4000
3	Protocol	TCP/IP-active, Specified	TCP/IP-passive, Specified
4	Send / Receive	Send / Receive	Send / Receive
5	Access point – IP address	192.168.0.2	192.168.0.1
6	Access point – Port No.	4000	4000
7	Send Timing	Cyclic sending: 1 second	Cyclic sending: 1 second
8	Transmission area	WR0 to WRF	WR10 to WR1F
9	Receiving area	WR10 to WR1F	WR0 to WRF

[Description of sample program]

The connection is established between EHV-CPU4 and Web controller in two seconds after RUN of EHV-CPU4, and the sending and receiving are started.

(If the connection is established before RUN of EHV-CPU4, the connection is closed immediately after RUN and the connection is established again.)

[Sample program of EHV-CPU4]



[Sample program of Web controller]

When Web controller is set to TCP/IP passive station, the connection stands by to open when the power supply is turned on or immediately after the connection is cut. Therefore a program to control the connection is unnecessary.

[Sample 4]

Network consists of two EHV-CPUs as follows, and the Control Editor sets the ASR communication. The setting information of two EHV-CPUs is as follows.

16-word data from WR0 to WRF of EHV-CPU5 is transmitted from EHV-CPU5 to EHV-CPU6. In EHV-CPU6, the received data is stored in WR0 to WRF.

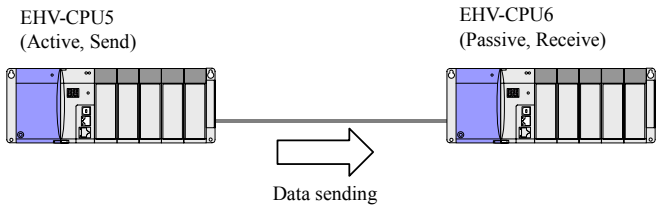


Figure 2.23 Connection diagram of Sample 4

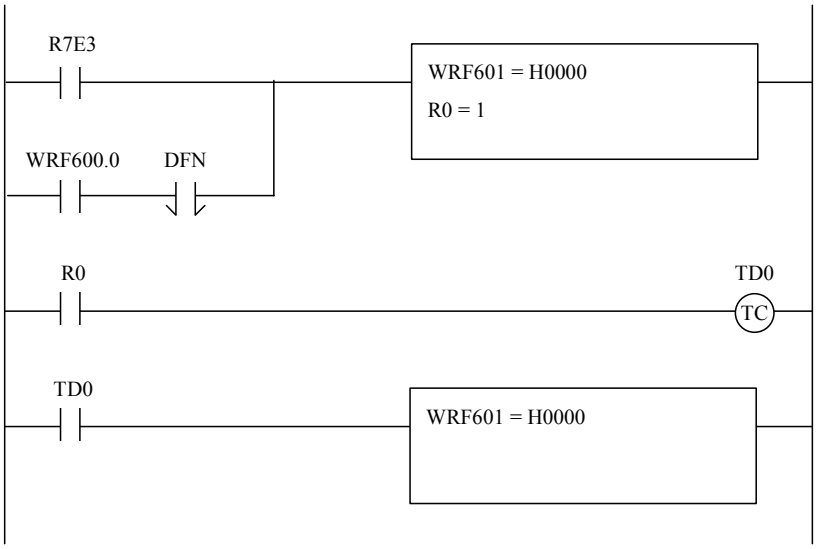
Table 2.13 Setting information of Sample 4

	Setting	EHV-CPU5	EHV-CPU6
1	IP address	192.168.0.101	192.168.0.102
2	Port No.	4000	4000
3	Protocol	UDP/IP, Specified	UDP/IP, Specified
4	Send / Receive	Send	Receive
5	Access points – IP address	192.168.0.102	192.168.0.101
6	Access points – Port No.	4000	4000
7	Send Timing	Cyclic sending: 40ms	—
8	Transmission area	WR0 to WRF	—
9	Receiving area	—	WR0 to WRF

[Description of sample program]

ASR port is opened in two seconds later after RUN of EHV-CPU5. If there is a communication target is on the network (there is a response to ARP packet), the transmission is executed automatically. And similarly, EHV-CPU6 executes the receiving if the ASR port is opened in EHV-CPU6 because the ASR port is opened also in two seconds after RUN of EHV-CPU6.

[Sample program of EHV-CPU5 and 6]



ASR port is closed immediately after RUN or in falling which ASR port status flag turns off after ASR port 1 is closed.

TD0 turns on in two seconds later after ASR port close.

ASR port open request bit is turned on in two seconds later after ASR port close, and ASR port is opened. In case of UDP/IP, it can be sent and received in this status.

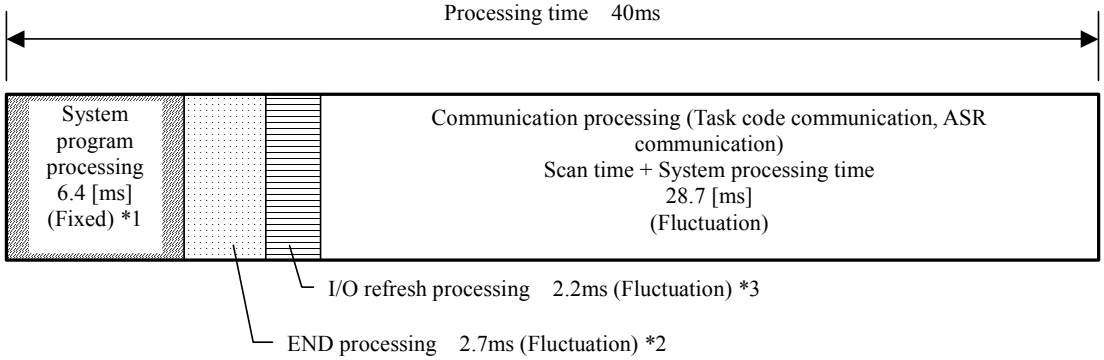
Cautionary note

In communication (Task code processing and ASR communication) of EHV-CPU series, the response may delay if CPU is loaded. We would explain a load of CPU below. Please utilize this as a standard in considering the responsibility of communications.

[About Communication processing time in CPU module]

CPU module executes processing by two processors, one is an operation processor to execute user programs and the other is a main processor to execute a system program processing, and END processing and a communication processing. While the operation processor executes the user program, the main processor executes the communication processing. Therefore, a total of scan time and system processing time is communication processing time.

Taking 40ms which is a minimum transmission interval of ASR communication for instance, a rate of time for each processing executed by the main processor is shown below. In this case, response delay and transmission delay occur at least unless task code communication and ASR communication are completed within approx. 28ms for communication processing.



*1. 1ms cyclic processing. In this case, it is executed 40 times.
*2. Processing time when the number of program steps is 0
*3. Processing time when the number of I/O mounted is 66 words (1056 points, that is 66 modules equipped 16-point I/O each are mounted.)

Figure 2.24 An example of ratio of each processing by main processor within 40ms

Communication processing time is not fixed but fluctuates under the influence of the following factors.

(a) Scan time of user program

While the operation processor executes the user program, the main processor can execute the system processing and the communication processing. The longer scan time becomes, the more communication processing time increases. And since the number of times that END processing occurs within a same time also decreases if scan time get longer, END processing decreases and communication processing time increases in above figure.

(b) I/O configuration

I/O refresh processing time increases and communication processing time increases in proportion to the number of modules mounted. It takes 1μs per word as processing time depending on the external I/O points and the number of link data.

(c) System processing time (Setting by Control Editor)

This is a setable time, using operation parameter of Control Editor. This “System processing” means the communication time.

[Calculation method of communication processing time]

Communication processing time can calculate using the following methods, and is calculated on the basis of the transmission interval of ASR.

(Example)

- Whole processing time (ASR transmission interval) … 40ms = 40,000μs
- System program processing time (1ms cyclic processing) … 160μs
- Scan time (Displayed value of CPU status current value) (*) … 2ms = 2,000μs
- END processing time (Fixed) (*) … 80μs
- I/O refresh processing time (In mounting 66 module with 16-point I/O) (*) … 66μs
- System processing time (Setting by Control Editor) … 1ms = 1,000μs

(*) Operation time for one scan

(a) System program processing time:

160 [μs/time] × 40 [time] (Execution times in 40ms) = 6,400[μs]

(b) Number of scan times of user program:

(Whole processing time - (a)) / (Scan time + END processing time + I/O refresh time + System processing time)

= (40,000[μs] - 6,400[μs]) / (2,000[μs] + 80[μs] + 66[μs] + 1,000[μs])

≅ 10.7 [time]

(c) Scan time:

2,000 [μs/time] × (b) = 2,000 [μs/time] ×10.7 [time] = 21,400 [μs]

(d) System processing time:

1,000 [μs/time] × (b) = 1,000 [μs/time] ×10.7 [time] = 10,700 [μs]

Communication processing time = (c) + (d) = 21,400 [μs] + 10,700 [μs] = 32,100 [μs] = 32.1ms

In this example, the communication processing time is 32.1ms. If processing of task code communication and ASR communication is 32.1ms or less, delay of communication response does not occur.

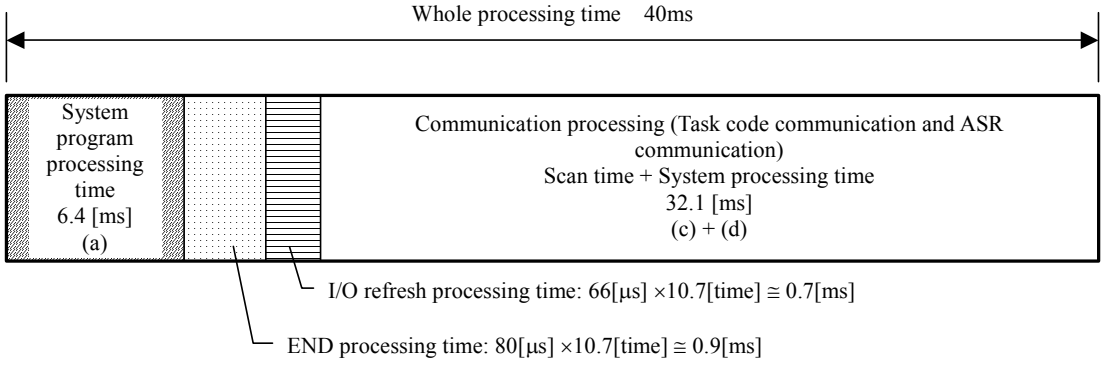


Figure 2.25 Calculation of communication processing time

[About Task code processing time]

In EHV-CPU, Task code communication is executed when Control Editor is connected in Online mode and the indicator such as a touch panel is connected. Processing time of task code changes according to I/O points to be monitored. A relation between I/O points to be monitored and task code is shown below. (Task code transmission interval is 110ms according to an original measurement conditions.)

Processing time of task code can be expressed as follows.

[Task code processing time] = 0.1 × [I/O monitor points] + 5.0 [ms]

But, when task code communication is executed using several ports, it takes processing time for used ports.

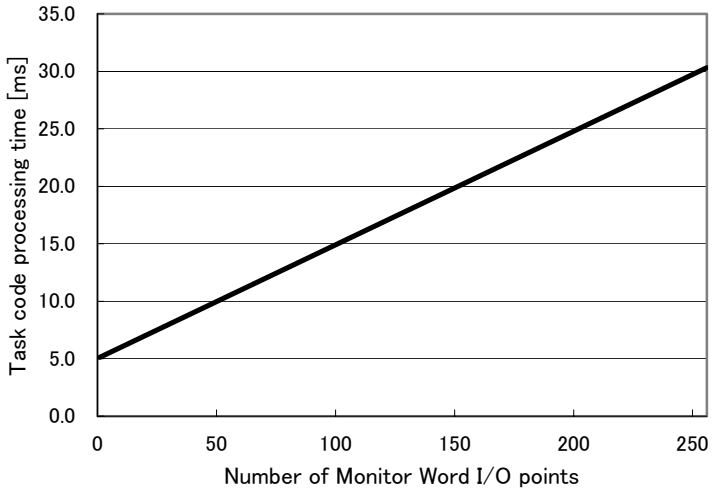


Figure 2.26 Monitor I/O points and Task code processing time

[About ASR communication time]

Processing time of ASR transmission depends on transmitted I/O points. A relation between transmitted I/O points and ASR processing time is shown below.

Processing time of ASR transmission is expressed as follows.

[ASR transmission processing time] = 0.0066 × [Number of transmitted words] + 3.7 [ms]

But, when ASR transmission is executed using several ports, it takes processing time for used ports.

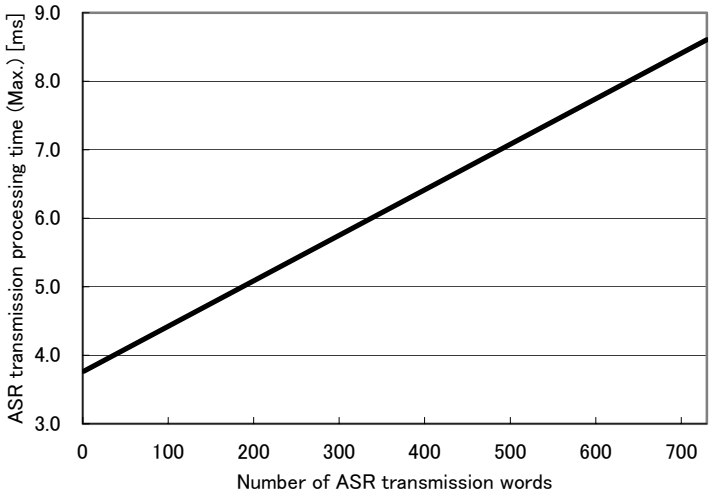


Figure 2.27 Monitor I/O points and Task code processing time

[About Transmission interval design of ASR communication]

ASR communication is set to the lower priority than Task code communication in communication processing. Therefore, the time which subtracts the task code processing time from the communication processing time is a time given to ASR communication. And if the processing of whole ASR communication is completed within this time, the set transmission interval is not kept and transmission delay will occur. Refer to the following examples in designing the transmission interval of ASR communication.

Example 1) ASR communication: Using 4 ports (each 32 words)

Task code port: using 1 port (each 32-word monitor)

[ASR transmission processing time] = (0.0066 × 32 [word] + 3.7) × 4 [port] = 15.6 [ms]

[Task code processing time] = (0.1 × 32 [word + 5.0) ×1 [port] = 8.2 [ms]

[Communication processing time] = [ASR transmission processing time] + [Task code processing time]= 15.6 + 8.2 = 23.8 [ms]

In the above case, the delay does not occur on transmission cycle in this condition because the time the main processor can be operated in the communication processing is 32.1ms.

Example 2) ASR communication: Using 1 port (256 words)

Task code port: Using 3 ports (each 32-word monitor)

[ASR transmission processing time] = (0.0066 × 256 [word] + 3.7) × 1[port] = 5.4[ms]

[Task code processing time] = (0.1 × 38[word] + 5.0) ×3[port] = 26.4[ms]

[Communication processing time] = [ASR transmission processing time] + [Task code processing time]= 5.4 + 26.4 = 31.8[ms]

In the above case, the delay may occur on transmission cycle occasionally because the time the main processor can be operated in the communication processing is 32.1ms.

Example 3) ASR communication: Using 4 ports (256 words)

Task code port: Using 2 ports (each 64-word monitor)

[ASR transmission processing time] = (0.0066 × 256[word] + 3.7) ×4[port] = 21.6[ms]

[Task code processing time] = (0.1 × 64[word] + 5.0) × 2[port] = 22.8[ms]

[Communication processing time] = [ASR transmission processing time] + [Task code processing time] = 21.6 + 22.8 = 44.4[ms]

In the above case, the delay occur on transmission cyclic in this condition because the time the main processor can be operated in the communication processing is 32.1ms.

[In order that the delay does not occur on communication processing]

According to the calculation methods in the preceding paragraph, when enough communication processing time cannot be obtained within 40ms cycle or when the expected communication response cannot be got, obtain the communication processing time by adjusting the following three items.

- The number of communication port to be used (ASR communication and Task code communication)
- The number of communication words
- System processing time

2.2.3 Reset function for Ethernet communication port

When communication using the Ethernet communication port cannot be performed by some factors, communication can be performed again by resetting the port. This port reset function can reset four task code ports and 6 ASR ports individually. Therefore, only a port to reset can be reset without stopping the port which is communicating normally.
But, a standard processing time for the port reset is 90ms.
Please turn ON a special internal output corresponding to the port to reset for using this function. The special internal output for this function is as follows.

Table 2.14 Special internal output for Ethernet communication port reset

No.	Name	Meaning	Description	Set condition	Reset condition
R910	Ethernet communication port Task code port 1 Reset request	1 : Reset request	Returns the processing of task code port 1 to the initial status.	ON by user	OFF by system (If reset is completed, OFF by system.)
R911	Ethernet communication port Task code port 2 Reset request	1 : Reset request	Returns the processing of task code port 2 to the initial status.		
R912	Ethernet communication port Task code port 3 Reset request	1 : Reset request	Returns the processing of task code port 3 to the initial status.		
R913	Ethernet communication port Task code port 4 Reset request	1 : Reset request	Returns the processing of task code port 4 to the initial status.		
R914	Ethernet communication port ASR port 1 Request reset	1 : Reset request	Returns the processing of ASR port 1 to the initial status.		
R915	Ethernet communication port ASR port 2 Request	1 : Reset request	Returns the processing of ASR port 2 to the initial status.		
R916	Ethernet communication port ASR port 3 Request reset	1 : Reset request	Returns the processing of ASR port 3 to the initial status.		
R917	Ethernet communication port ASR port 4 Request reset	1 : Reset request	Returns the processing of ASR port 4 to the initial status.		
R918	Ethernet communication port ASR port 5 Request reset	1 : Reset request	Returns the processing of ASR port 5 to the initial status.		
R919	Ethernet communication port ASR port 6 Request	1 : Reset request	Returns the processing of ASR port 6 to the initial status.		

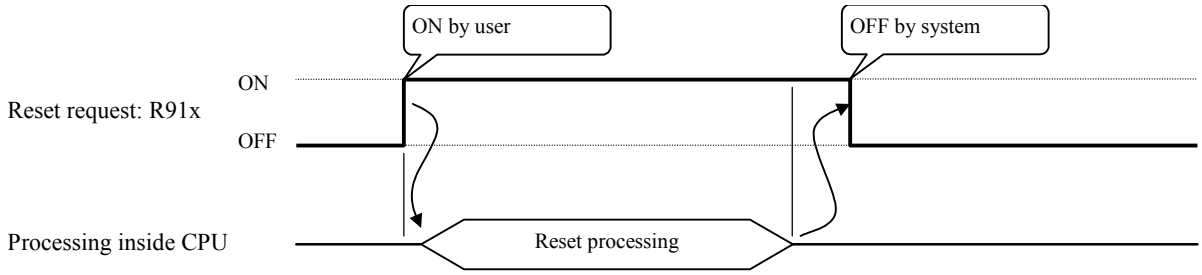


Figure 2.28 Reset function for Ethernet communication port

■ Cautionary note

The above bit can return the Ethernet communication processing inside EHV-CPU to the initial status but cannot reset the hardware.

And even if the reset request is transmitted by turning on the above bit of the task code port and the ASR port which cannot be used, it cannot be reset. Therefore, the reset request bit turned on by user is turned off by system.

Reference

The operation of reset processing is different between “a case where the connection is established” and “a case where the connection is not established”.

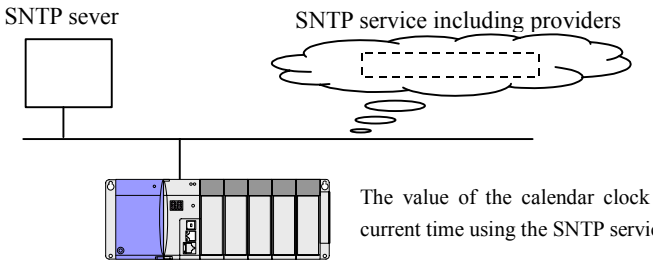
In the case where the connection is established, the connection is closed first (the timeout is performed at 1ms if the processing is not completed properly) and then the communication end inside EHV-CPU is deleted and re-constructed.

In the case where the connection is not established, the communication end inside EHV-CPU is deleted and re-constructed.

2.2.4 NTP client function

EHV-CPU is equipped with the SNTP (Simple Network Time Protocol) client function which retrieves the current time from the NTP (Network Time Protocol) server and the SNTP server on the network.

The interval of retrievals can be set by specifying time and minute. And this function can control when the current time is retrieved from a user program.



The value of the calendar clock in EHV-CPU can be synchronized by retrieving the current time using the SNTP service on intranet or Internet.

Figure 2.29 SNTP client function

Table 2.15 SNTP client specifications

Item	Specifications
Communication protocol	SNTP（Simple Network Time Protocol）
Retrieval interval	Specified by user（00:01～99:59）
Retrieved time data	yyyy/mm/dd/tt/mm/ss
Where the time data is stored	Special internal output area（WRF00B～WRF00F）
Update of clock using retrieved time	Updated by software in EHV-CPU

The current time retrieved with this function is stored on the Calendar clock are (WRF00B to WRF00F), and update is performed at the specified interval. And the current time retrieved from the NTP server is stored in the special internal output of EHV-CPU.

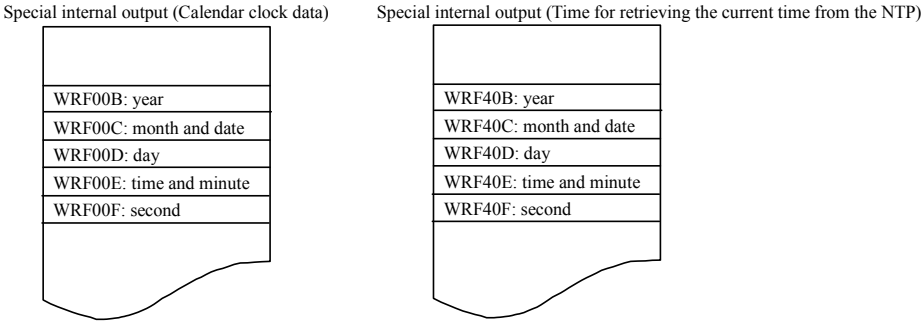


Figure 2.30 Word Special internal output for NTP function

Setup

The setup for the NTP function can be specified using the Control Editor.

Table 2.16 Setup for NTP function

	Item	Description
1	Valid	Specifies the setup for NTP function to valid or invalid.
2	Server address	Specifies the address of NTP server.
3	Connection interval	Specifies the interval for retrieving the current time from NTP server.
4	Time zone	Specifies the time zone.

It is controllable using the bit special internal output from the user program when the NTP server is accessed. And the setup of the time zone and the current time retrieved from the NTP server are stored in the word special internal output.

(1) Bit special internal output

Table 2.17 Bit special internal output for NTP function

No.	Name	Meaning	Description	Set condition	Reset condition
R900	NTP time retrieving User program Control Valid/Invalid	0: Invalid 1: Valid	Changes the timing when the current time from the NTP server is retrieved, by a cycle from CE or by a user program (R901).	ON by user	OFF by user
R901	NTP time retrieving request	1: Retrieval start	ON when the current time is retrieved from NTP server.	ON by user	OFF by system
R902	NTP time retrieving result	0: Success of retrieving 1: Failure of retrieving	Indicates the failure of retrieving the current time from NTP server.	ON by system	OFF by system

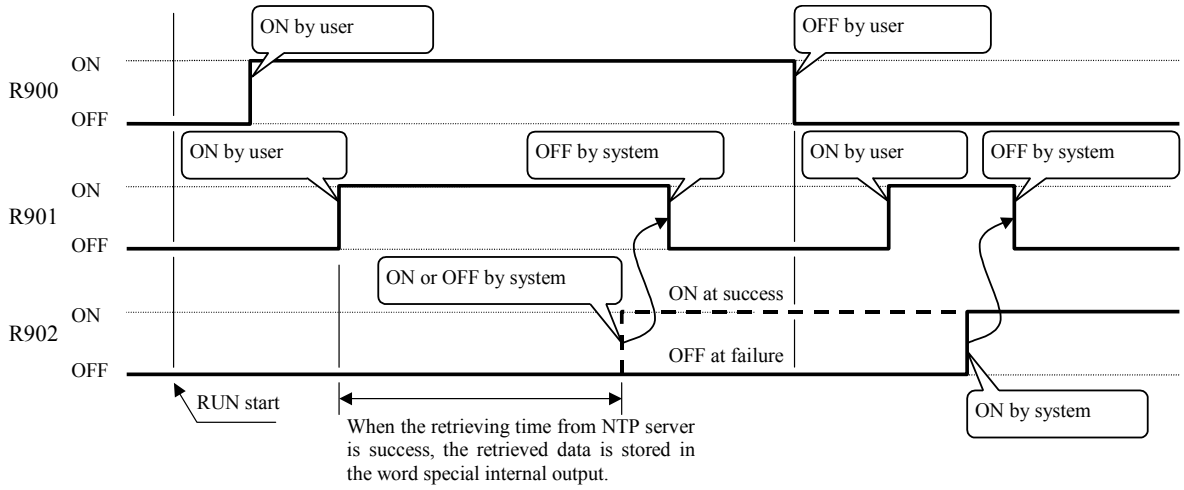


Figure 2.31 Bit special internal output for NTP function (when NTP setup is valid)

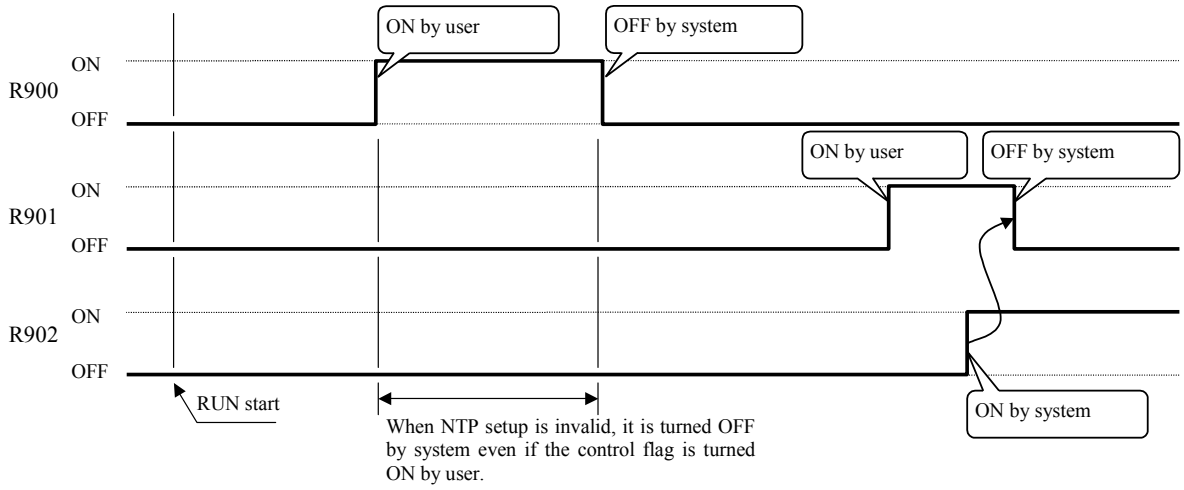


Figure 2.32 Bit special internal output for NTP function (when NTP setup is invalid)

■ Cautionary note

When the NTP time retrieving request (R901) competes with the read (R7F8) and the write (R7F9) of the calendar clock, and the 30s adjust (R7FA), another processing is not performed until the processing to the first detected request is completed because the first request detected by system is processed first.

(2) Word special internal output

Table 2.18 Word special internal output for NTP function

No.	Name	Storage data	Description	Set condition	Reset condition
WRF40A	Time zone setup	See the following table	Specifies the time zone of the NTP setup of EHV-CPU. But stores the time zone set value by system when the power is on.	Sets by user. (Sets by system only when the power is on.)	—
WRF40B	Calendar, Clock NTP server retrieving value (BCD 4-digit)	Year	Indicates year with 4-digit.	Sets by system. (Sets the current time before the time zone revision at success of the time retrieving from the NTP server.)	—
WRF40C		Month and date	Indicates month and date.		
WRF40D		Day	Indicates day. (Sun.: 0000 – Sat.: 0006)		
WRF40E		Time and minute	Indicates time and minute. (24 hours system)		
WRF40F		Second	Indicates second. (Lower is 2 digits, Upper is 00)		

Table 2.19 Time zone setup for NTP function

Set value	Time zone
H0000	GMT –12:00
H0001	GMT –11:00
H0002	GMT –10:00
H0003	GMT –9:00
H0004	GMT –8:00
H0005	GMT –7:00
H0006	GMT –6:00
H0007	GMT –5:00
H0008	GMT –4:00
H0009	GMT –3:30
H000A	GMT –3:00
H000B	GMT –2:00
H000C	GMT –1:00

Set value	Time zone
H000D	GMT
H000E	GMT +1:00
H000F	GMT +2:00
H0010	GMT +3:00
H0011	GMT +3:30
H0012	GMT +4:00
H0013	GMT +4:30
H0014	GMT +5:00
H0015	GMT +5:30
H0016	GMT +5:45
H0017	GMT +6:00
H0018	GMT +6:30
H0019	GMT +7:00

Set value	Time zone
H001A	GMT +8:00
H001B	GMT +9:00
H001C	GMT +9:30
H001D	GMT +10:00
H001E	GMT +11:00
H001F	GMT +12:00
H0020	GMT +13:00
Outside range	GMT

■ Cautionary note

When changing the time zone by the special internal output, the clock data becomes the data after revision after having seta time zone.

(The set value of time zone is stored on the backup memory. Note that the life of the backup memory gets shorter if the time zone setup is changed frequently.)

2.2.5 Factory setting

The factory setting of the Ethernet communication port is as following.

Table 2.21 Factory setting for Ethernet communication port

Item		Setup
IP address		192.168.0.1
Subnet mask		255.255.255.0
Default gateway		0.0.0.0
Ethernet communication speed (*)		10M half-duplex
NTP setting		Invalid
Time zone		GMT+09:00
Task code communication setting		
	Port 1	Valid
	Port 1 Port No.	3004
	Port 1 Protocol	TCP/IP
	Port 2	Valid
	Port 2 Port No.	3005
	Port 2 Protocol	TCP/IP
	Port 3	Valid
	Port 3 Port No.	3006
	Port 3 Protocol	TCP/IP
	Port 4	Valid
	Port 4 Port No.	3007
	Port 4 Protocol	TCP/IP
	Timeout	30
ASR setting		
	ASR port 0	Invalid
	ASR port 1	Invalid
	ASR port 2	Invalid
	ASR port 3	Invalid
	ASR port 4	Invalid
	ASR port 5	Invalid

(*) Since the Ethernet communication speed can be changed in Ver.x107 or newer, 5 types of communication speed (Auto-negotiation, 100M full-duplex/half-duplex, 10M full-duplex/half-duplex) can be set. The communication speed is set to 10M half-duplex at the shipment. (As for “x” in Ver.x107, it represents EHV-CPU128 when x is 0, it represents EHV-CPU64 when x is 1, it represents EHV-CPU32 when x is 2, and it represents EHV-CPU16 when x is 3. This version information is stored in the special internal output WRF050.)

Control Editor (Ver.2.00 or newer) or IP address setting tool can change the Ethernet communication speed.

2.3 Serial communication port

2.3.1 Pin arrangement

(1) RS-232C

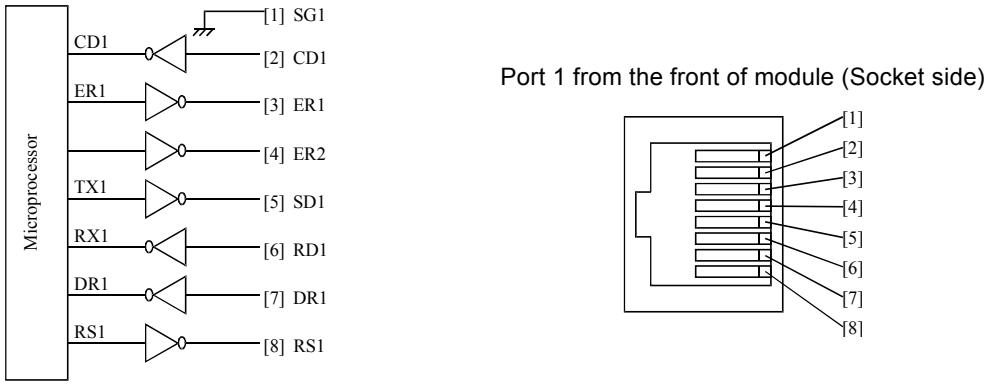


Figure 2.33 Circuit diagram and pin numbers for Serial communication port RS-232C

Table 2.21 List of signals for Serial communication port RS-232C

Pin No.	Signal abbreviation	Direction CPU : Host	Meaning
[1]	SG1	⟷	Grand for signal
[2]	CD1	←	Notification signal during carrier received
[3]	ER1	→	Communication enabled signal. When this signal is high level, communication is possible.
[4]	ER2	→	Outputs High.
[5]	SD1	→	Data transmitted by CPU
[6]	RD1	←	Data received by CPU
[7]	DR1	←	Peripheral units connected signal. When this signal is high level, indicates that dedicated peripheral are connected.
[8]	RS1	→	Transmission request signal. When this signal is high level, indicates that the CPU can receive data.

(2) RS-422 / 485

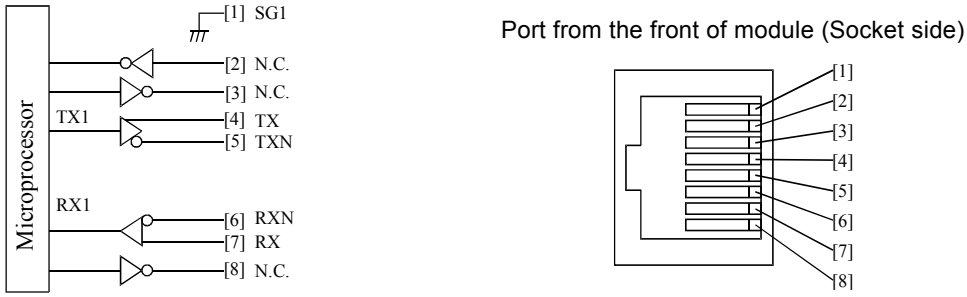


Figure 2.34 Circuit diagram and pin numbers for Serial communication port RS-422 / 485

Table 2.22 List of signals for Serial communication port RS-422 / 485

Pin No.	Signal Abbreviation	Direction CPU : Host	Meaning
[1]	SG	⟷	Grand for signal
[2]	N.C.		Un used. Do not connect.
[3]	N.C.		Un used. Do not connect.
[4]	TX	→	CPU transmission data +
[5]	TXN	→	CPU transmission data -
[6]	RXN	←	CPU receiving data -
[7]	RX	←	CPU receiving data +
[8]	N.C.		Un used. Do not connect.

(3) Pin arrangement for RS-422 / 485

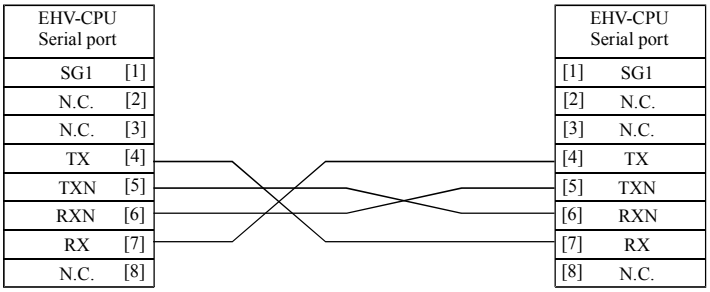


Figure 2.35 Signal connection diagram for RS-422

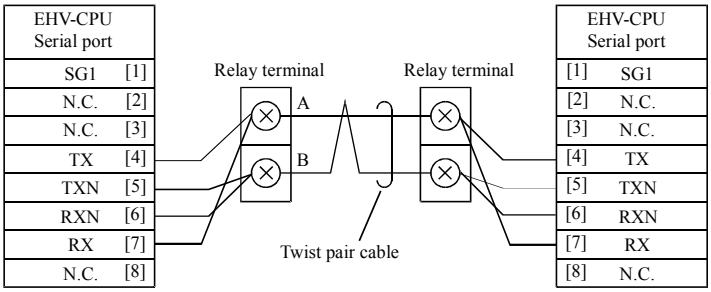


Figure 2.36 Signal connection diagram for RS-485

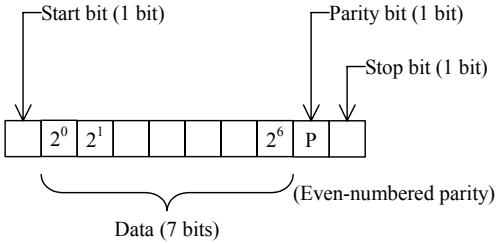
2.3.2 Dedicated port

The specifications in setting the serial communication port as a dedicated port is shown in the table 2.21.

In the dedicated port, a CPU program can be created or monitored from the programming device connected. Also, a monitoring system which uses a monitor available on the market can be constructed. Moreover, a variety of systems can be constructed by connecting a personal computer and creating software.

For setup and setting the connecting cables, please check beforehand whether it is used as the purpose.

Table 2.23 Specifications for a dedicated port

Item	Specification		
Transmission speed *1	4,800 bps、9,600 bps、19,200 bps、38,400 bps、57,600 bps		
Interface *1	RS-232C	RS-422	RS-485
Maximum cable length	15 m	500 m	500 m
Connection mode (Maximum connected units)	1 : 1	1 : 1 / 1 : N (32 units)	1 : 1 / 1 : N (32 units)
Communication system	Half duplex system		
Synchronization system	Start-stop synchronization system		
Startup system	One-side startup system using the host side command		
Transmission system	Serial transmission (Bit serial transmission)		
Transmission code	ASCII		
Transmission code configuration			
Transmission code outgoing sequence	out from the lowest bit in character units.		
Error control	Vertical parity check, Sum check, Overrun check, Framing check		
Transmission unit	Message unit (variable length)		
Maximum message length	1,460 bytes (including control characters)		
Control procedure *1	H-series dedicated procedure (High protocol) Standard procedure 1 (Transmission control procedure 1), Simple procedure (Transmission control procedure 2) *2		
Connector used	8-pin modular connector (RJ-45 type)		

*1 Communication speed, communication interface, and control procedure are set using a Control Editor.
It is set to 1:1 for the transmission control procedure 1, RS-232C, and 38,400 bps at shipment.
The setup becomes effective when the power supply turns on next.

*2 Transmission control procedure 2 is a simple communication protocol not support communication via CPU link network.

[Specifications of RS-422 and 485 of Serial communication]

Serial communication system of EHV-CPU is Half duplex. Half duplex is a communication system which can transmit to only one direction of both directions communication. When the serial communication port for EHC=CPU is used as a dedicated port, EHV-CPU repeats an operation of responding the request transmitted from the peripheral device. Until the start of the receiving processing from the response end, the delay may occur. It is a communication error because the receiving is not performed properly if the request is transmitted before the receiving is started.

For the safety communication, leave an interval of 2ms between the response end and the receiving processing start.

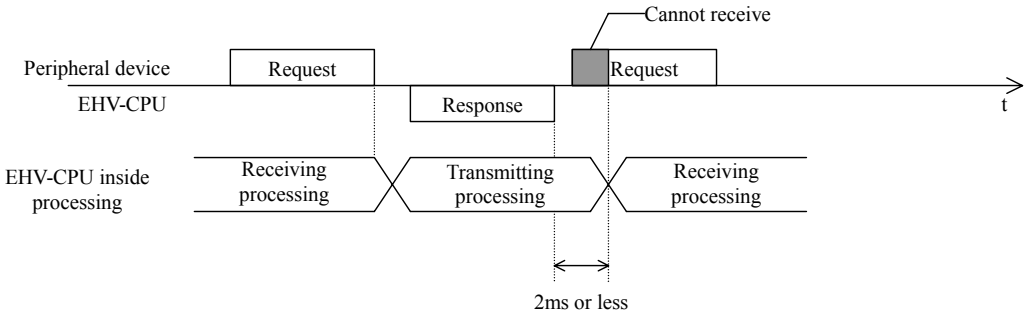


Figure 2.37 Communication specifications of RS-422/485 of Serial communication

[Example]

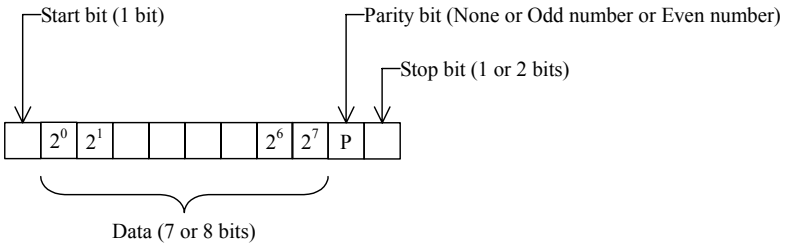
In case of the serial communication to HMI (Human machine Interface) such as a touch panel and an indicator, set the transmission wait on HMI when communication error occurs frequently in EHV-CPU.

2.3.3 General-purpose port

Serial communication port can be used as a general-purpose port. When it is specified as a general-purpose port, the transmitting and receiving operations are controlled by the user program. The specifications for the general-purpose port is shown in the table 2.24.

For setup and setting of the connected cables, please check beforehand whether it is user as the purpose.

Table 2.24 Specifications for a general-purpose port

Item	Specification		
Communication speed	300 bps、600 bps、1,200 bps、2,400 bps、4,800 bps、9,600 bps、19,200 bps、38,400 bps、57,600 bps		
Interface	RS-232C	RS-422	RS-485
Maximum cable length	15 m	500 m	500 m
Connection mode (Maximum connected units)	1 : 1	1 : 1 / 1 : N (32 units)	1 : 1 / 1 : N (32 units)
Communication system	Half duplex system		
Synchronization system	Start-stop synchronization system		
Startup system	One-side startup system using the host side command		
Transmission system	Serial transmission (Bit serial transmission)		
Transmission code	Definition by user		
Transmission code configuration			
Transmission code outgoing sequence	out from the lowest bit in character units		
Error control	Vertical parity check, Overrun check, Framing check		
Transmission unit	Message unit (variable length)		
Maximum message length	1,024 bytes (including control characters)		
Control procedure	No procedure		
Control code	Definition by user		
Connector used	8-pin modular connector (RJ-45 type)		

* The setting information becomes effective when the power supply turns on next.

(1) 1: N communication (RS-485)

(A) Precautions

When performing 1 to N communication using RS-485, communicate in polling/selecting mode. When creating a ladder program, note the following points.

[1] Communicate by making sure the master station and slave station are using the same start code.

[2] The master station should transmit a request by specifying the station number of the slave station.

[3] The slave station should transmit a response only when the request from the master station is to the own station. Set the station so that it will reset the mode and wail for the next request in the event is received a request addressed to other station.

[4] The master station should transmit a new request after at least 20 ms (t_p in figure below) has elapsed from the time it completed receiving the last response from the slave station.

[5] The slave station should transmit a request after at least 20 ms (t_s in figure below) has elapsed from the time it completed receiving the request from the master station.

An example of 1 to N transmission sequence is shown below. This example shows a sequence in which the master station transmits a series of requests to the slave stations 1 to 3, and a sequence in which the slave station received the request transmits the response.

In Fig.2.28, a salient expressed in solid line indicates that the station received a transmission addressed to the own station, and a salient expressed in dotted line indicates that the station received a transmission addressed to other station.

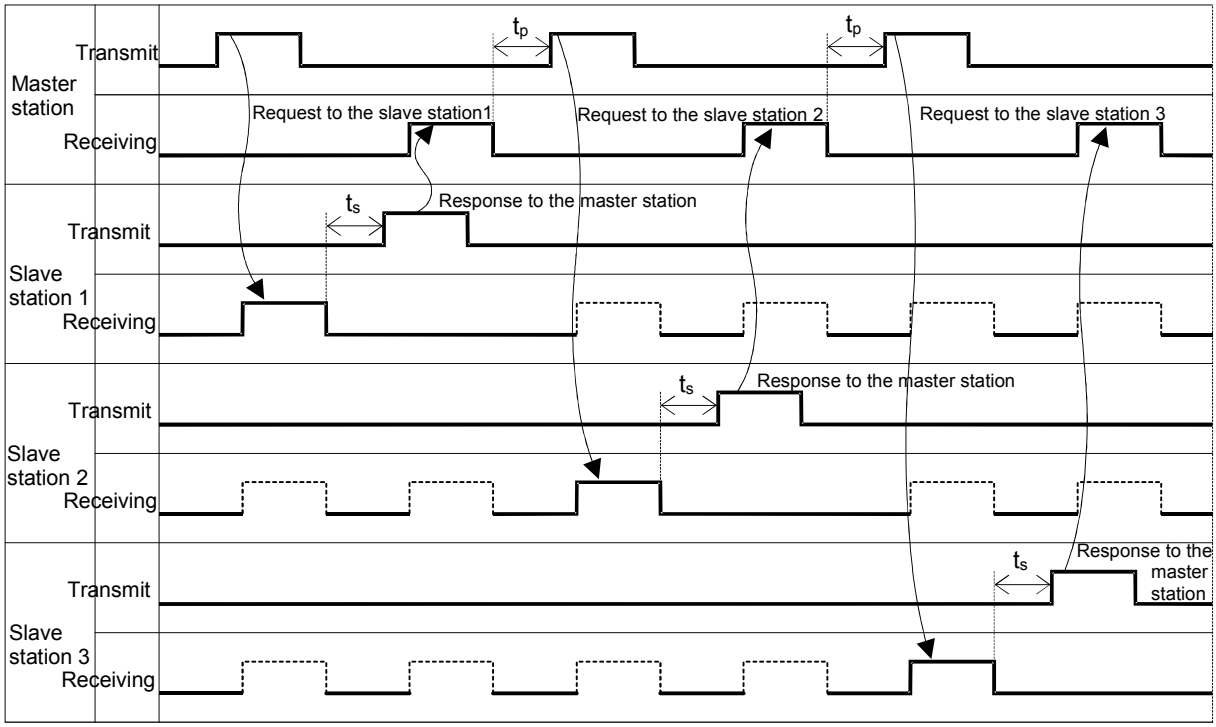


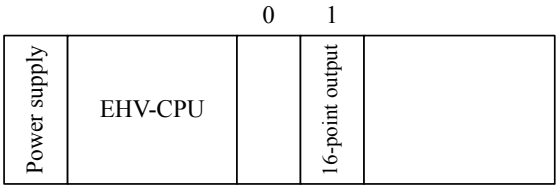
Figure 2.37 1 : N Transmitting and receiving sequence

(B) Sample program

The following shows a simple program which communicates between one master station and three slave stations using RS-485.

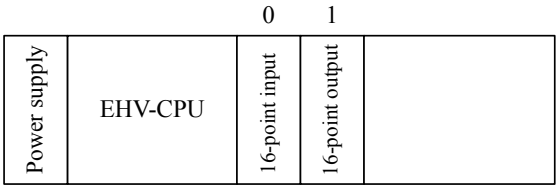
[1] Mounting the module

(a) Master station side



Mounts the 16-point output module in the slot 1 of the basic base.

(b) Slave station side



Mounts the 16-point input module in the slot 0 of the basic base, and the 16-point output module in the slog 1 of the basic base.

[2] Assigning internal output

A sample program is created using the following assignments. In actual cases, change the I/O number etc. according to the application.

(a) Assigning internal output in the master station side

I/O	No.	Usage
WM	100 to 10E	TRNS 0 command Parameter area (s to s+14)
R	000 to 00B	TRNS 0 command Communication control bit area (t to t+11)
	100	Transmission data setting completion flag
WR	0000 to 001F	Transmission data area (32-word)
	0100 to 011F	Receiving data area (32-word)
	4000	Slave station number for communication
	4001	Number of slave stations
WL	001 to 003	Storage area for receiving data

(b) Assignment internal output in the slave station

I/O	No.	Usage
WM	0000 to 000E	RECV 0 command Parameter area（s to s+14）
	200 to 21F	Transmission data area（32-word）
	300 to 31F	Receiving data area（32-word）
WR	0000 to 000E	TRNS 0 command Parameter area（s to s+14）
	0200 to 021F	Transmission data area（32-word）
	0300 to 031F	Receiving data area（32-word）
	4000	Slave station number during communication
	4001	Own station number
WL	001 to 003	Storage area for receiving data

[3] Transmission format

Transmission formats between the master station and slave stations are as follows.

(a) Request format from the master station to the slave station. (A maximum of 3 bytes)

Start code	Slave station No.	End code
02H	1 to 3	0DH

(b) Response format from the slave station to the master station. (A maximum of 5 bytes)

Start code	Own station No.	Data		End code
02H	1to 3	Any*	Any*	0DH

*: Any data can be set except the end code (0DH). The slave station number is set in this sample program.

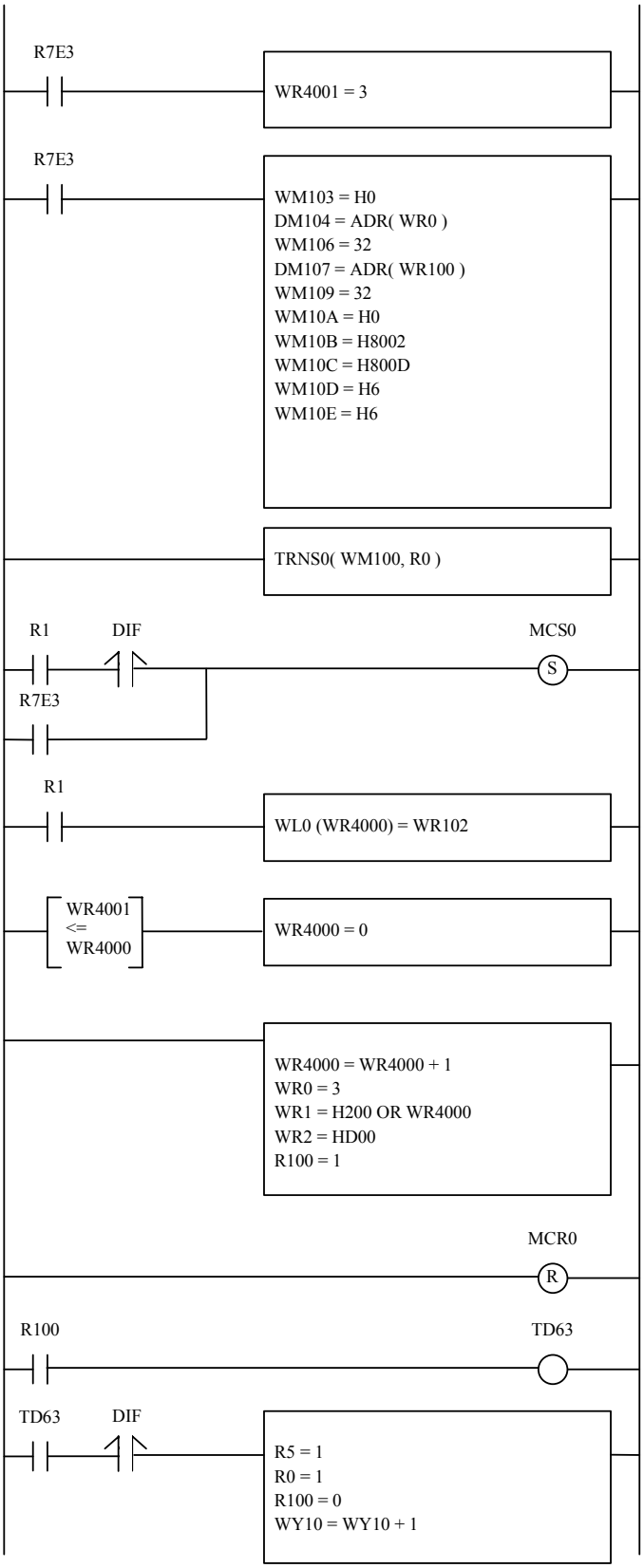
[4] Receiving result in the master station side

If the transmission between the slave stations 1 to 3 complete successfully, the following data is set in the WL area of the master station. The slave station sets its own slave station number as part of the data.

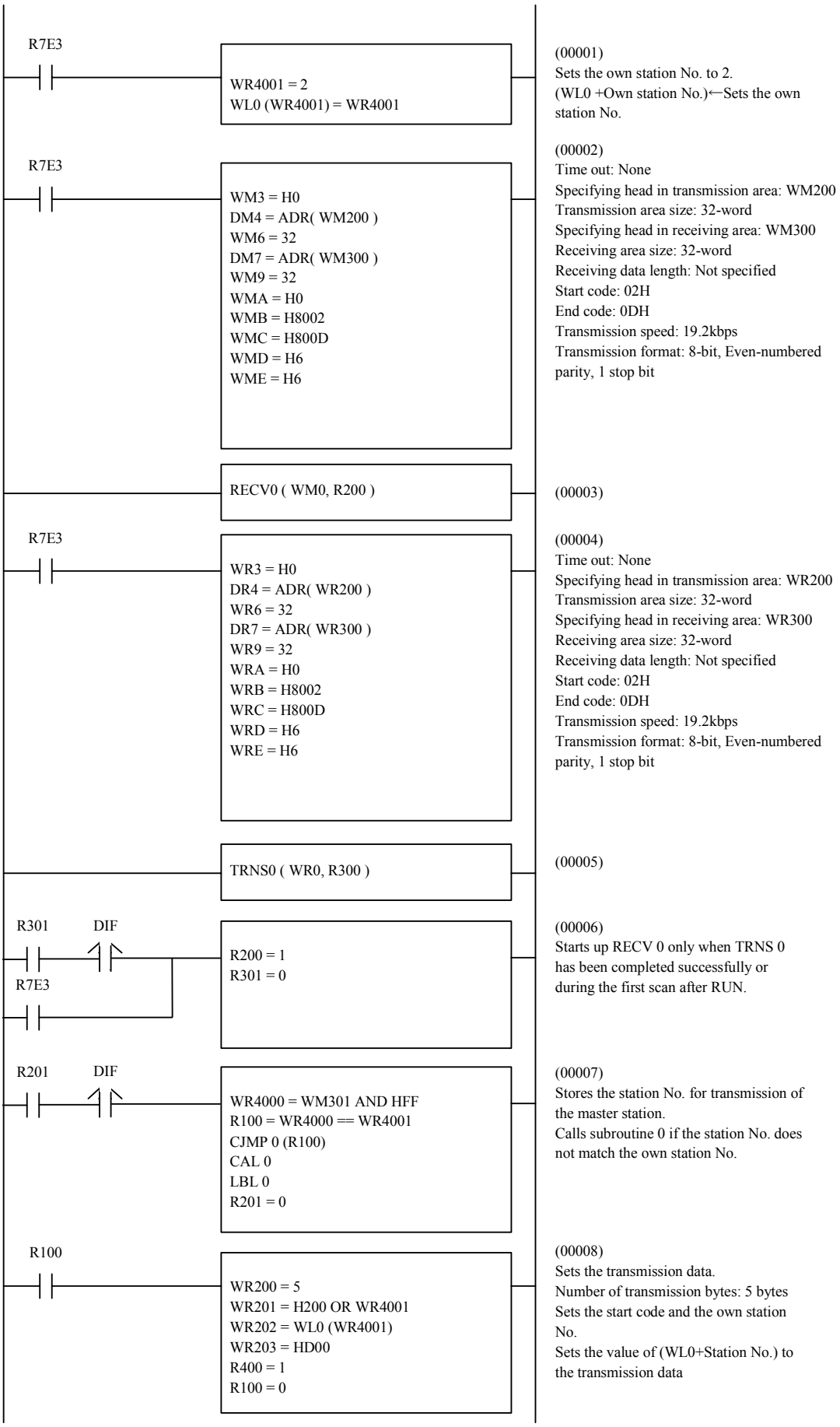
Address	Set value	Description
WL0001	0001H	Received data from the slave station 1.
WL0002	0002H	Received data from the slave station 2.
WL0003	0003H	Received data from the slave station 3.

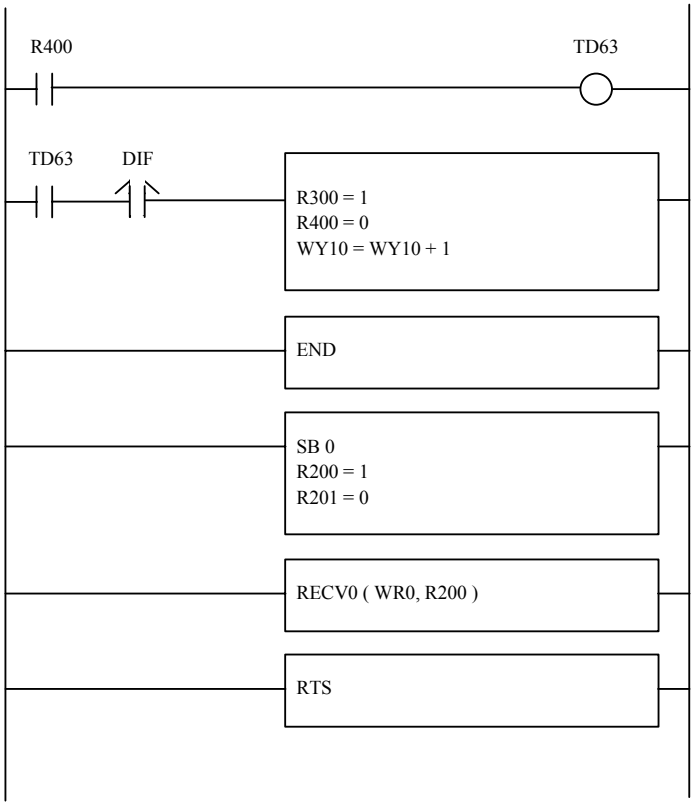
[5] Program

(a) Program on the master station side (with three slave stations)



(b) Program in the slave station side (slave station No. 2)




R400
TD63
(00009)
1ms 20
TD63
DIF
R300 = 1
R400 = 0
WY10 = WY10 + 1
(00010)
Starts up TRNS 0 after 20 ms has been elapsed.
END
(00011)
SB 0
R200 = 1
R201 = 0
(00012)
Subroutine 0
Turns on the receiving execution flag.
Turns off the receiving normal completion flag.
RECV0 (WR0, R200)
(00013)
RTS
(00014)

2.3.4 Modem connection function

EHV-CPU is equipped with a model connection function. The model connection function can be controlled using task codes. The setup on the Control Editor is needed to use this function.

Refer to the following table for communication specifications.

If a difference of communication speed between two operating modems, connecting between them may be difficult. Therefore, please combine to eliminate any difference in communication speeds.

(1) Configuration

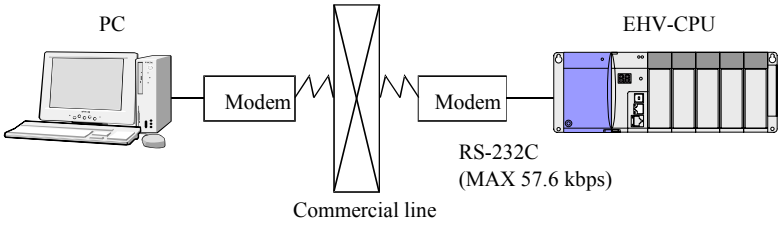
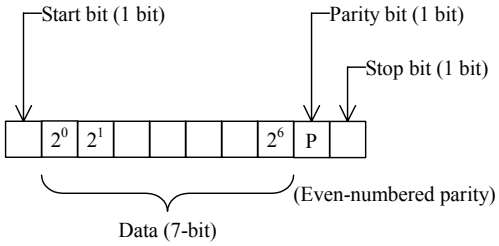


Figure 2.38 Connection configuration of modem

(2) Specifications

Table 2.25 Specification for Modem connection function

Item	Specification
Communication speed	2,400 bps、4,800 bps、9,600 bps、19,200 bps、38,400 bps、57,600 bps
Communication system	Full duplex system (Communication program is half-duplex control.)
Synchronization system	Start-stop synchronization system
Transmission system	Serial transmission (Bit serial transmission)
Transmission code	ASCII code
Transmission code configuration	
Transmission code outgoing sequence	out from the lowest bit (20) in character units.
Error detection	Vertical parity check, Overrun check, Framing check
Interface	Conforms to RS-232C
Control procedure	H-series dedicated procedure (High protocol)
Startup system	One-side startup system by the host side command
Time out detection at connecting modem	Sets by Control Editor
Time out detection at communicating modem	Sets by Control Editor

* ER signal cannot be controlled. Therefore, the line is cut by commands or control by connecting the line is needed using other I/O.

Table 2.26 List of signals for Serial communication port at connecting modem

Pin No.	Signal abbreviation	Direction CPU Modem	Meaning
1]	SG1	⟷	Grand for signal
2]	CD1	←	Notification signal during carrier received. Connects to CD in the modem.
3]	ER1	→	Communication enable signal of the terminal
4]	ER2	→	Unused
5]	SD1	→	Data transmitted by CPU. Connects to SD in the modem.
6]	RD1	←	Data received by CPU. Connects to RD in the modem.
7]	DR1	←	Communication enable signal of the modem. Connects to DR in the modem.
8]	RS1	→	Transmission request signal. Connects to RS in the model.

(3) AT command

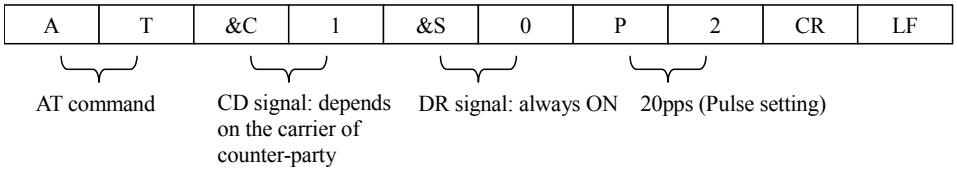
AT command is used for setting various setup of a modem and set using the host computer. EHV-CPU generates the AT command for the initial setting automatically. The AT command is not used for other purpose.

Refer to the instruction manual of the modem maker for the AT command.

In AT commands, an instruction transmitted to the modem from the host is called “command” and the character string in response to the “command” returned to the host from the mode is called “result code”.

At commands always begins the character string “AT” and a return code is input at the end of the command. However, A/ is excluded. The command that follows the “AT” can have multiple inputs in a single line.

Example)



(A) Format

[1] AT command format

A	T	Command	Parameter	Command	Parameter	⋯	CR	LF

[2] Result code format

CR	LF	Result code (word)	CR	LF

Result code (number)	CR	LF

(B) List of commands（extract）

[1] AT command

Command	Function overview		Example
AT	Automatically recognizes data format.		—
A/	Re-executes the response directly preceding.		—
ATA	Forced receiving		
ATDmm	Dial		ATD12345678
ATEn	Command error (echo-back the input character string in the modem)	0: Excluding 1: Including	ATE0
ATHn	Line ON/OFF	0: On hook (disconnect) 1: On hook	ATH0 ATH1
ATPn	Pulse setting (dial)	0,1: 10 pps 2: 20 pps	ATP0、ATP1 ATP2
ATQn	Setting of result code excluding/including	0: Including 1: Excluding	ATQ0
ATT	Tone setting (push)		ATT
ATSn=X	Sets S register value.		ATS0=0
ATVn	Result code display format	0: Number 1: Word	ATV0 ATV1
AT&Cn	CD signal control	0: Always ON 1: Depends on the carrier of counter-party modem	AT&C0 AT&C1
AT&Dn	ER signal control	0: Always ON 2: Line disconnection by turning from ON to OFF during communication 3: Resets software by turning from ON to OFF	AT&D0 AT&D2 AT&D3
AT&Sn	DR signal	0: Always ON 1: Depends on sequence 2: Depends on CD signal	AT&S0 AT&S1 AT&S2
AT&Rn	RI(CI) signal control	0: ON from calling start till communication start 1: ON from calling start till communication end 2: ON/OFF in synchronization with the call signal	AT&R0 AT&R1 AT&R2

[2] S register

S register	Set value	Function
S0	0 No automatic receiving 1 to 255	Setting for automatic receiving / receiving ring count
S2	0 to 127 (43 [+])	Escape code setting
S3	0 to 127 (13 [CR])	CR code setting
S4	0 to 127 (10 [LF])	LF code setting

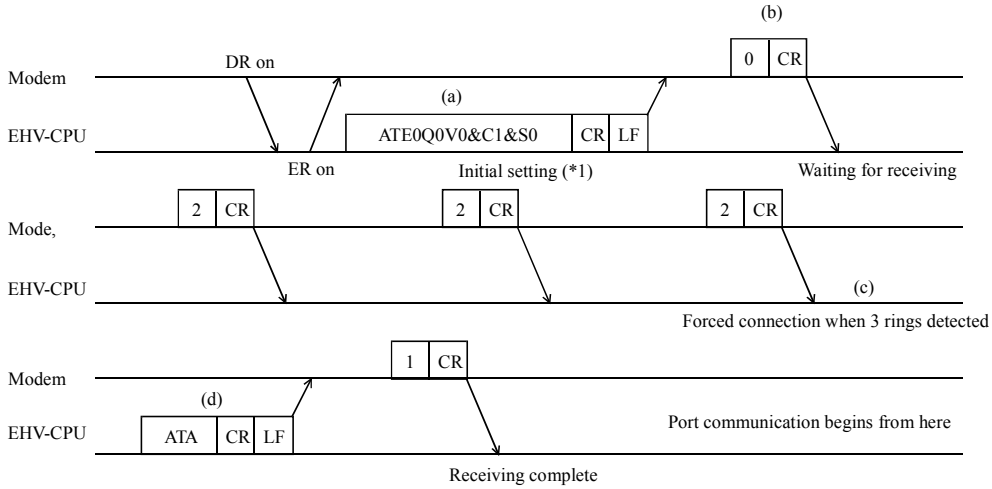
[3] Result code

Number format	Word format	Meaning
0	OK	Normal execution
1	CONNECT	Connection complete
2	RING	Receiving detection
3	NO CARRIER	Line disconnection
4	ERROR	Command error
5	CONNECT 1200	1200 bps Connection
6	NO DIAL TONE	Cannot hear dial tone
7	BUSY	Busy signal detected
8	NO ANSWER	No tone heard
10	CONNECT 2400	2400 bps Connection
11	CONNECT 4800	4800 bps Connection
12	CONNECT 9600	9600 bps Connection
13	CONNECT 14400	14400 bps Connection

(C) Sequence

An example of a communication sequence using the Omron-made modem ME3314A is shown below.

[1] Receiving sequence



(a) The PLC generates the AT command for performing the initial setting.

(b) If the initial setting is OK, the modem returns “0”.

(c) The PLC detects the result code “2” three times in the status waiting for receiving.

(d) Connects the modem.

[2] Disconnect sequence

(a) The PLC disconnects the line if the result code “3” is returned form the modem.

*1: The initial setting for modem by EHV-CPU sets the minimal items as follows. Therefore, please set the modem by the AT command connecting a personal computer with the modem before connecting with EHV-CPU. (Set the DR signal to always ON.) However, do not change the following initial setting.

Initial setting contents

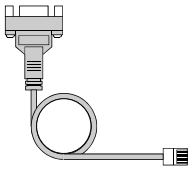
Command echo	: Excluding
Result code	: Including
Result code display forma	: Number format

*2: Please generate a task code (H1C) of the disconnected request from the host side before disconnecting the line in practice.

2.3.5 Connection between Serial communication port and Peripheral device

Table 2.27 shows cables to connect a peripheral unit to RS-232C interface for serial communication port of EHV-CPU.

Table 2.27 Peripheral unit connection configuration

<table>
<tr><th>Peripheral unit</th><th colspan="2">Calbe</th></tr>
<tr><td rowspan="2">Programming software
Control Editor
  </td><td>WVCB02H
  </td><td>EH-RS05
  </td></tr>
<tr><td colspan="2">EH-VCB02
  </td></tr>
</table>

2.3.6 Connection method for RS-422 / 485 communication

Serial communication port of EHV-CPU can communicate with an interface of RS-422 / 485. Communication of 1:N stations can be performed using H-series dedicated control procedure (high-protocol) or a general-purpose procedure with a general-purpose port command (TRNS 0, RECV 0). Figure 2.30 and 2.31 show examples when a connection is made for 1:N stations.

And the connection for the communication in 1:1 made is the connection of only the first EHV-CPU.

(1) In case of RS-422

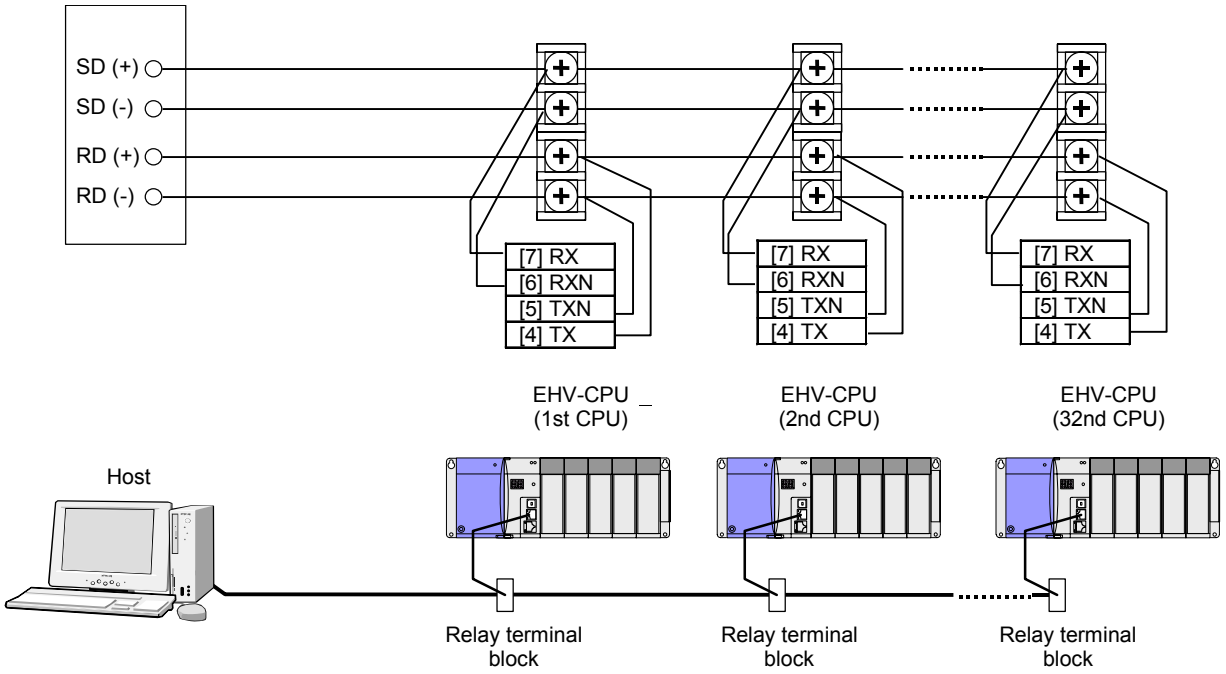


Figure 2.39 Connection for 1:N station communication by RS-422

(2) In case of RS-485

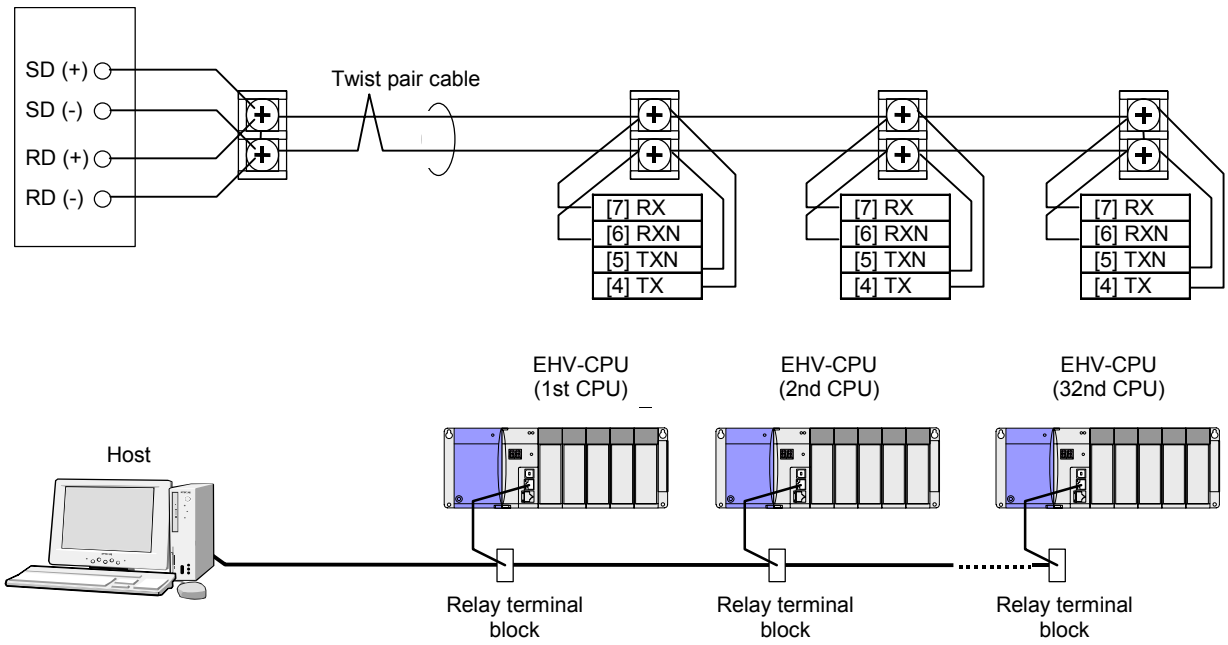


Figure 2.40 Connection for 1:N station communication by RS-485

2.4 USB communication port

USB communication port of EHV-CPU supports USB2.0. (Transfer speed is up to 12Mbps in FULL Speed.)

The USB communication port is a dedicated port for connecting the Control Editor. It can be programmed and monitored. Table 2.28 shows specifications for the USB communication port.

Table 2.28 Specifications for USB communication port

Item	Specification
Standard	Conforms to USB2.0
Transfer speed	FULL Speed (Maximum 12Mbps)
Communication protocol	For Control Editor connection

⚠ Caution

If it is connected using the Control Editor and the USB communication port, the communication error may occur on the Control Editor in noise environment. Please connect using serial port or LAN port if the communication error occurs in noise environment. And for a stable communication, do not bring a communication cable close to other wiring, and do not put the cable and the wiring in the same duct.

Appendix 1 Cable Connection Diagram

A cable connection diagram in case of connecting peripheral devices with EHV-CPU with a RS-232C interface of a serial port is shown below.

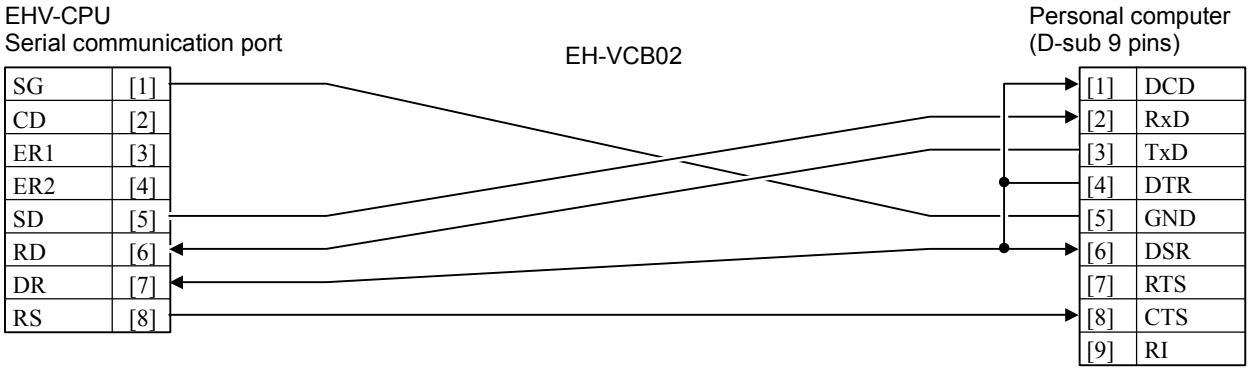


Figure A-1.1 EH-VCB02 connection diagram

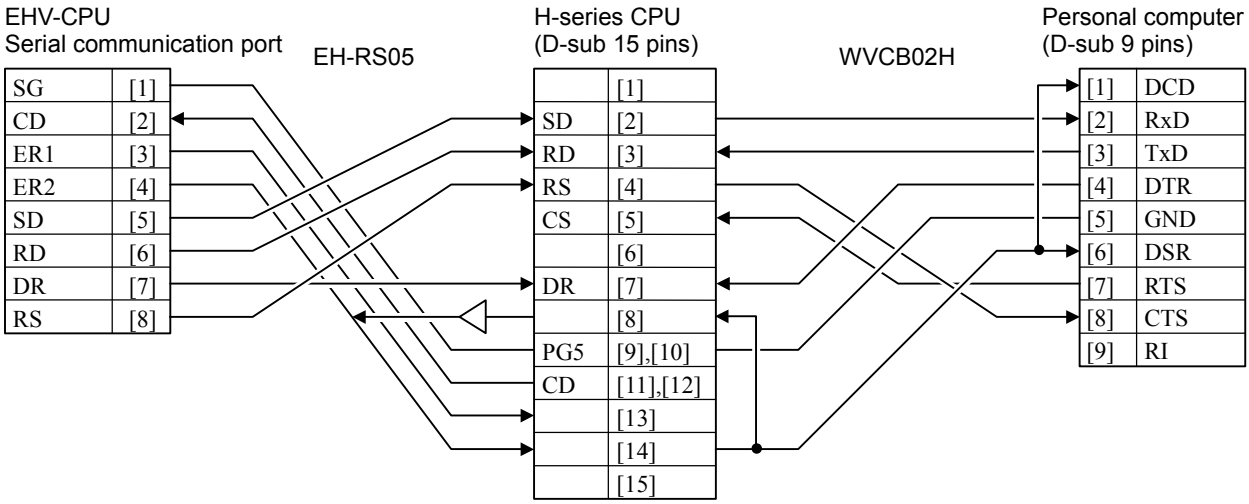


Figure A-1.2 EH-RS05 + WVCB02H connection diagram